Delivering Performance on Sun: System Tuning

Technical White Paper

Greg Schmitz and Allan Esclamado

Sun microsystems

Contents

Introduction 1

This document focuses on techniques for tuning the Sun computing environment for maximum performance. It is aimed at system administrators and sophisticated endusers who do their own system administration. Each chapter concentrates on a different subsystem of the computing environment and the specific things that can be done to increase performance. The subsystem topics are presented in the order recommended for tuning analysis and optimization:

- Chapter 2 — Tuning the Solaris Kernel — identifies the parameters that can be tuned to ensure the Solaris operating environment operates at peak efficiency.
- Chapter 3 — Memory— discusses the impact of paging and swapping schemes, and identifies tools administrators can use to effectively monitor and manage memory and swap space.
- Chapter 4 — Tuning Disk Subsystems — focuses on the key performance criteria, important disk technologies, and data layout schemes that ensure high performance I/O. Sun tools that can aid in the detection, isolation, and correction of performance-related issues are also discussed.
- Chapter 5 — File System Tuning — looks at the high performance file systems available from Sun, as well as the tools and techniques that can be employed to ensure that these file systems continue to operate at peak efficiency.

- Chapter 6 focuses on the networking technologies available today, as well as the issues surrounding choosing a networking technology, ensuring data integrity, evaluating network performance, and reducing network congestion. Tools available to aid these efforts on Sun systems are also discussed.
- Chapter 7 highlights the major functions of the Solaris kernel, and discusses the key tables and environment variables that need to be tuned in order to ensure it operates efficiently.

Other Performance White Papers and Sun Documentation

This paper, as well as companion documents, can be found at *http://www.sun.com* on the World Wide Web. These documents include:

- *Delivering Performance on Sun: Optimizing Applications for Solaris* helps software developers construct high performance applications with the use of proper compiler flags. It also discusses the development of high performance multithreaded applications.
- *Delivering Performance on Sun: Tools for High Performance Distributed Computing* addresses performance issues for distributed computing platforms and the compute-intensive applications they run.

In addition, all Sun documentation can be found at *http://docs.sun.com*.

Tuning the Solaris Kernel 2

The Solaris operating system, called the *kernel*, is loaded into memory when the system boots. The kernel manages all of the physical resources of the computing system. Applications interface with the kernel through a series of system calls.

Key kernel functions include:

- Implementing many types of file systems (UFS, NFS, tmpfs, etc.) and permitting processes to create, read, write, and access these files.
- Managing daemons, system processes which manage system resources.
- Moving large groups of information, called *pages*, to and from the disk into main memory as needed. This concept is known as *virtual memory*.
- Keeping track of all active processes and deciding which one gets to run next.
- Managing *device drivers*, the programs that control physical devices like disk drive subsystems.
- Managing network services and IPC facilities.
- Providing system management functions, such as booting and error handling.
- Providing other functions that make system resources available to processes.

Reconfiguring the Solaris Kernel

The standard Solaris kernel is dynamic, supporting all devices, pseudo-devices, and options by automatically loading modules when they are needed. Unlike previous versions of SunOS, there is no need to manually configure the Solaris kernel.

The kernel needs to be reconfigured after hardware has been installed, and a reconfiguration reboot (`boot -r`) performed before it can be recognized and used by the system. For some configuration changes it is sufficient to run the `drvconfig` command followed by the `disks` or `tapes` command to recognize devices recently added to the system.

The /etc/system File

The /etc/system file houses a wide variety of system configuration and tuning parameters. Administrators can tune many of the parameters contained in /etc/system to increase system performance, including buffer cache, file system, STREAMS, interprocess communication, memory management, and other parameters.

- *bufhwm*, the maximum size for buffer cache memory usage, in KB
- *ufs_ninode*, the maximum size of the inode table
- *ncsize*, the size of the directory name lookup cache
- *nstrpush*, the maximum number of STREAMS pushes allowed
- *strmsgsz*, the maximum size for a STREAMS message a user can create
- *strctlsz*, the maximum size of the ctl part of a message
- *strthresh*, the maximum amount of dynamic memory the STREAMS subsystem can consume, in bytes
- *sadcnt*, the number of `sad` devices
- *msginfo_msgmap*, the number of entries in the message map
- *msginfo_msgmax*, the maximum message size
- *msginfo_msgmnb*, the maximum bytes on queue
- *msginfo_msgmni*, the number of message queue identifiers
- *msginfo_msgssz*, th segment size of a message (should be a multiple of the word size)
- *msginfo_msgtql*, the number of system message headers
- *msginfo_msgseg*, the number of message segments (must be < 32768)
- *seminfo_semmap*, the number of entries in the semaphore map
- *seminfo_semmni*, the number of semaphore identifiers
- *seminfo_semmns*, the number of semaphores in the system

- *seminfo_semmnu*, the number of processes using the undo facility
- *seminfo_semmsl*, the maximum number of semaphores, per id
- *seminfo_semopm*, the maximum number of operations, per semaphore call
- *seminfo_semume*, the maximum number of undo structures per process
- *seminfo_semvmx*, the semaphore maximum value
- *seminfo_semaem*, the maximum value for adjustment on exit
- *shminfo_shmmax*, the maximum shared memory segment size (`set shmsys:shminfo_shmmax = 0xffffffff,` or `set shmsys:shminfo_shmmax = 4292967295`)
- *shminfo_shmmin*, the minimum shared memory segment size
- *shminfo_shmmni*, the number of shared memory identifiers
- *shminfo_shmseg*, the segments, per process
- *lotsfree*, if freemem drops below lotsfree, the system starts to steal pages from processes
- *tune_t_fsflushr*, the rate at which fsflush is run, in seconds
- *tune_t_minarmem*, the minimum available resident (not swappable) memory needed to avoid deadlock, in pages
- *tune_t_minasmem*, the minimum available swappable memory needed to avoid deadlock, in pages
- *tune_t_flckrec*, the maximum number of active frlocks
- *lwp_default_stksize*, the size of the kernel stack for lwps. This value should not be adjusted unless there is a kernel overflow. lwp_default_stksize is expressed in bytes and must be a multiple of PAGESIZE bytes.
- *npty*, the total number of 4.0 or 4.1 pseudo-ttys configured
- *pt_cnt*, the total number of 5.7 pseudo-ttys configured
- *priority_paging*, enabling the sytem to place a boundary around the file cache, ensuring that file system I/O does not cause application paging
- *consistent_coloring*, identifies the method used to improve hit ratios for the CPU external cache — 0(uses various virtual address bits), 1 (physical address is set to virtual address), 2 (bin hopping), 6 (SBSD page coloring scheme)

Administrators can modify these kernel parameters while the system is running via the `adb` command, or by editing the /etc/system file and rebooting the system.

More information /etc/system tunable parameters can be found in the *System Administration Volume II* manual located at *http://docs.sun.com* on the World Wide Web.

Kernel Tables

A variety of kernel tables can be tuned by system administrators to enhance the performance of the Solaris operating environment. Table 2-1 identifies these tables and the relevant environment variables they control, as well as suggested settings for each.

	Variable	Default Setting
Process Table	max_procs	20 + 16 * maxusers
Per User Process Limit	maxuprc	max_nprocs - 5
Inode Cache	ufs_ninode	max_nprocs + 16 + maxuers + 64
Quota Table	ndquot	(maxusers * NMOUNT) / 4 + max_nprocs
Directory Name Lookup Cache	ncsize	max_nprocs + 16 + maxusers + 64

Table 2-1 System administrators can tune key Solaris kernel tables

Many of the traditional Unix kernel limits have been removed over the years. Solaris now dynamically sets limits that increase on large memory configurations. Traditional tables that no longer exist or need to be tuned include the text table, open file table, callout table, and clist table.

More information on configuring kernel tables can be found in *Sun Performance and Tuning, Java and the Internet*, *Chapter 14: Kernel Algorithms and Tuning*.

maxusers Kernel Parameter

Over time, the Solaris kernel has been modified to enable it to scale according to the hardware capabilities of the workload it is running. By self-configuring and tuning, Solaris reduces administrative burdens and dynamically adapts to changing business needs. While earlier versions needed to be hand-tuned, each successive version of Solaris 2 has converted hand-adjusted values into adaptively changing limits.

One kernel parameter that remains an important factor in system performance is the `maxusers` environment variable. As evidenced in Table 2-1, many Solaris kernel parameters are defined relative `maxusers`, rendering it perhaps the most critical variable administrators can tune. Indeed, `maxusers` determines whether or not the Solaris kernel has sufficient room to accommodate users comfortably. (It is important to note that `maxusers` is not a limit on the number of simultaneous users supported by the system.)

Increasing the `maxusers` parameter increases the size of the Solaris kernel, and consequently decreases the amount of memory available. Administrators should, therefore, follow a few simple guidelines to ensure the kernel is tuned effectively for users of the system:

- The `maxusers` variable is automatically set by Solaris according to the amount of physical memory available in the system. The automatic minimum value for `maxusers` is 8, and the automatic maximum is 1024.
- The `maxusers` parameters is set a little less than the amount of RAM in the system, in megabytes.
- For NFS servers with small amounts of memory (under 512 MB), it can be beneficial to increase the DNLC and inode cache limits.

There is no need to tune in most cases and only a small impact on performance can be expected from any kernel tuning.

Additional information on the impact of the `maxusers` kernel parameter can be found in *Sun Performance and Tuning, Java and the Internet, Chapter 14: Kernel Algorithms and Tuning*.

Memory 3

Memory Paging and Swapping

When memory becomes tight, the kernel begins to employ techniques known as *paging* and *swapping*. Paging occurs when the kernel puts part of an executing program out to disk, enabling the pages it occupied to be used for other purposes. When the process needs to access the page of memory that was moved to disk, a *page fault* occurs, forcing the process to wait until the page is copied back into memory. This scheme fosters the efficient use of memory, as only the parts of a program that are in use at the time the program is running need to be in memory and available for use.

Swapping occurs when the kernel stores an entire currently inactive process to disk, enabling the pages it occupied to be used by other executing programs. To continue the process is paged back in gradually. Swapping is a part of the normal housekeeping performed by the Solaris kernel.

Paging and Swapping Parameters

The Solaris kernel utilizes several values stored in the /etc/system file to determine when to start paging and swapping, actions that begin and end when the memory system crosses a threshold set as part of the kernel paging algorithm:

- *lotsfree*, the amount of memory required by the page daemon. If memory drops below this limit, the page daemon starts looking for pages to reuse.
- *desfree*, the level at which swapping begins.

- *minfree*, the minimum memory level acceptable to the system.
- *slowscan*, the number of pages scanned per second when *lotsfree* is reached.
- *fastscan*, the number of pages scanned per second when *minfree* is reached.
- *handspreadpages*, the distance between the “hand” that sets a reference flag and the “hand” that clears it. The two “hands” keep track of which pages of memory have been accessed recently. If the pages have not been accessed in a certain amount of time, they are paged out.
- *maxpgio*, the maximum number of page I/O operations per second the system will schedule. *maxpgio* is typically set to 40, assuming there is little swap space on multiple devices. If system swap space is on fast and/or multiple disks, *maxpgio* should be increased to take advantage of the higher performance devices.

More information on paging in the Solaris kernel can be found in *Sun Performance and Tuning, Java and the Internet, Chapter 13: RAM and Virtual Memory*. Information on the /etc/system file can be found in the *System Administration Guide, Volume II* located at *http://docs.sun.com* on the World Wide Web.

Processes and Paging

Understanding the number of processes running on a system and how they are affected by paging is essential to overall performance. System administrators should routinely monitor processes and learn how they are utilizing system resources. To aid this effort, Sun has provided several tools for administrators as part of the Solaris operating environment: `vmstat,` swap, sar, `ps` and Find Process. Each tool can be used in a variety of ways, including determining memory, paging, and swapping measurements.

Memory Measurements

The first step in evaluating processes is determining the paging and swapping activity occurring on the system. Both the `vmstat` and `sar` commands can be used to monitor paging activity. While `sar` is better for information logging purposes, `vmstat` tends to provide more detailed information. Correlations exist between the output of these tools, as discussed below.

- Swap space (`vmstat swap`, `sar -r freeswap`, and `swap -s`)

  Each command describes the amount of available swap space, although in different forms. The `vmstat swap` command identifies the amount of available swap space, in KB while the `sar -r freeswap` reports in 512 byte blocks. The swap -s command shows available swap space in addition to other metrics. Administrators should note that when available swap space is exhausted the system will be unable to use additional memory.

- Free memory (`vmstat free` and `sar -r freemem`)

  Administrators can determine the amount of free memory — the pages of RAM that are ready immediately to be used when a process starts execution or needs more memory. The `vmstat free` command reports free memory in KB, the while `sar -r freemem` command reports it in pages.

- Reclaims (`vmstat re`)

  The number of pages reclaimed from the free memory list are called *reclaims*. Pages that have been taken from a process but not yet reused by another can be reclaimed, thereby avoiding the full page fault that would typically result. The `vmstat re` command can be used to identify reclaimed memory.

Page Fault Measurements

Identifying when page faults are occurring is an important step in evaluating system performance.

- Attaches to existing pages (`vmstat at` and `sar -p atch`)

  The `vmstat at` and `sar -p atch` commands measure the number of attaches to shared memory pages.

- Pages paged in (`vmstat pi` and `sar -p pgin, ppgin`)

  The `vmstat pi` command reports the number of KB/second, while the `sar` command reports the number of page faults and the number of pages paged in by swap space for file system read operations. Administrators should note that the file system block size is 8 KB. Consequently, there may be two pages, or 8 KB, paged in per page fault on machines with 4 KB pages. Also noteworthy is the fact that UltraSPARC machines employ 8 KB pages; almost all other CPUs utilize 4 KB pages.

- Minor faults (`vmstat mf` and `sar -p vflt`)

  Minor page faults are caused by address space or hardware address translation faults that can be resolved without performing a page-in operation. Such operations are fast and require little CPU utlization, eliminating the need for a process to have to stop and wait.

- Other fault types (`sar -p pflt`, `slock`, `vmstat -s copy-on-write`, `zero fill`)

  Many other types of page fault conditions are possible, and can be identified using the above commands. More information on these and other fault conditions can be found in the corresponding man pages, as well as *Sun Performance and Tuning, Java and the Internet, Chapter 13: RAM and Virtual Memory*.

Page-out Measurements

The `vmstat` and `sar` commands provide information on the amount of data paged out to the swap space or filesystem:

- Pages paged out (`vmstat po` and `sar -g pgout, ppgout`)

  The `vmstat po` command reports the number of KB/second page out, while `sar` reports the number of page-outs and the number of pages paged out to swap or file system space. Administrators should note that the clustering of swap space writes can lead to a large number of pages written per page-out.

- Pages freed (`vmstat fr` and `sar -g pgfree`)

  The Solaris operating environment keeps track of the pages freed — the rate at which memory is returned to the free list. The `vmstat fr` command reports the amount of memory freed in KB freed per second, while the `sar` command reports the number of pages freed per second.

- Short-term memory deficit (`vmstat de`)

  Deficit is a paging parameter that provides a measure of hysteresis for the page scanner when there is a period of high memory demand. If the value returned by the vmstat de command is non-zero, then memory was being consumed quickly and extra free memory will be reclaimed in anticipation of its impending use.

- Page daemon scanning rate (`vmstat sr` and `sar -g pgscan`)

  The `vmstat sr` and `sar -g pgscan` commands can be used to determine the number of pages scanned by the page daemon as it looks for pages used infrequently. If the page daemon scanning rate stays above 200 pages per second for long periods of time, then a memory shortage is likely.

- Pages freed by the inode cache

  Statistics related to the inode cache used by the Unix file system (UFS) are also indicators of system performance. In particular, ufs_ipf measures UFS inode cache reuse, a feature that can cause pages to be freed when an inactive inode is freed. ufs_ipf is the number of inodes with pages freed as a percentage of the total number of inodes freed.

Swapping Measurements

The `sar` and `vmstat` commands also supply system swapping information.

- Pages swapped in (`vmstat -S si` and `sar -w swpin,bswin`)

  The `vmstat -S si` command reports the number of KB/second swapped in, while the `sar -w swpin` command reports the number of swap-in operations. The `sar -w bswin` command reports the number of 512 byte blocks swapped in.

- The `vmstat -S so` command reports the number of KB/second swapped out, while the `sar -w swpot` command reports the number of 512 byte blocks swapped out.

More information on analyzing paging statistics can be found in *Sun Performance Tuning, Java and the Internet, Chapter 13*.

Managing Swap Space — The `swap` Command

The efficient operation of a system relies on the effective utilization of both memory and swap space. System administrators must ensure the system has sufficient swap resources so the paging and swapping activities performed by the kernel can operate smoothly. Sun provides the `swap` utility in the Solaris operating environment to facilitate the addition, deletion, and monitoring of the swap areas used by the memory manager.

- *Adding a swap area*

  The `-a` option of the `swap` command can be used by system administrators to add a swap area to the system. A swap file must be specified, such as /dev/dsk/c0t0d0s1, as well as a length for the swap area in 512 byte blocks. A swap space must be at last 16 blocks in length.

- *Deleting a swap area*

  The `-d` option of the `swap` command can be used by system administrators to delete a swap area from the system. Once deleted, blocks will no longer be allocated from this area. In addition, all swap blocks previously in use in this swap area will have been moved to other swap areas.

- *Listing swap area status*

  The `-l` option of the `swap` command can be used by system administrators to list the status of all swap areas on the system. Several pertinent pieces of information are reported, including: the path name for the swap area, the major/minor device number if a block special device is employed, the length of the swap area, and the available free space.

- *Printing summary information*

  The `-s` option of the `swap` command can be used by system administrators to summary information for all swap areas on the system. Total swap space usage and availability is provided. System administrators can determine the total amount of swap space allocated, the amount not currently allocated but reserved for future use, the total amount that is either allocated or reserved, and the amount available for future reservation and allocation.

Distributing Swap Space

To achieve peak system performance, system administrators must distribute swap space so it can be utilized effectively. Following a few simple guidelines can dramatically improve system performance:

- Place swap partitions or swap files on as many different disks as possible.
- Place swap partitions or swap files on the fastest disks in the system.
- If more than one disk is available per controller, limit the swap space to only one of those disks.

- Although swapping to remote (NFS) file systems is permitted, this practice should be avoided if at all possible. Swapping to remote file systems not only degrades swap performance, it also increases network traffic.
- Avoid placing swap files on file systems that are suspected of excess fragmentation or are almost full. If possible, swap files should be placed in a dedicated disk partition to eliminate fragmentation concerns.

More information on managing swap space can be found in the `swap(1M)` man page.

Process Manager

The new Process Manager in Solaris 7 enables administrators to quickly obtain process related information in the CDE environment (Figure 3-1). The Process Manager enables administrators to:

- Show user, system or all processes
- Sort processes by resource consumption
- Suspend, resume, or kill processes
- Diagnose sick processes and contact their owning user via Address Manager

More information on the Process Manager can be found in the Solaris operating environment administrative guides located at *http://docs.sun.com* on the World Wide Web.

Obtaining Process Information—ps

System administrators need to obtain information about active processes in the system. The `ps` command aids this effort by reporting the process ID, terminal identifier, cumulative execution time, and the command name by default. More information, including processes owned by a specific user, can be obtained by utilizing the options specified in the `ps(1M)` man page.

Address Space Map

The `/usr/proc/bin/pmap` command can be used to print the address space map for each running process. Furthermore, the new `-x` option available in the Solaris 7 operating environment prints resident/shared/private mapping details, providing more information than ever before.

More information on the pmap command can be found in the `proc(1M)` man page.

Tuning Disk Subsystems 4

This chapter first describes options for selecting disk subsystems which may improve system performance. Following is a description of the tools and utilities for analyzing the performance of these disk subsystems.

It is not unusual for storage capacity requirements to grow by as much as 100% per year. Coincident with this growth is an increasing need to ensure ready access to data whenever it is needed. System administrators know that raw capacity and performance do not stand alone — both scalability and high availability are important practical requirements for ensuring economical and reliable operation and information access. Indeed, disk subsystems must be configured and tuned to match the performance, availability, and cost requirements of users.

Focusing on disk subsystems, this chapter describes key performance criteria, introduces important disk technologies, and describes Sun tools that can aid in the detection, isolation, and correction of performance-related issues. References for more detailed tuning information are provided throughout.

Disk Performance Specifications

A number of factors are helpful in determining the relative performance of disk subsystems to one another:

- *Disk capacity*, the amount of data that can be stored on the disk or in the disk subsystem.

- *Seek time*, the amount of time it takes to move the disk drive's heads from one track to another, and is a product of the performance of the actuator. Disk drives typically spend ten times more time seeking than transferring data, making seek time perhaps the most important indicator of a disk's performance. Some disk vendors quote the time it takes to traverse the entire disk; others cite the time it takes to traverse one-third of the disk.
- *Rotational speed*, the speed at which the disk spins. After the head has moved to the appropriate track, the disk must rotate until the data is underneath the head. The disk's rotational speed determines its latency—the amount of time it takes to set up for a data transfer.
- *Data transfer rate*, the amount of data that can be transferred from the disk to the host per unit time. Both theoretical maximum and typical transfer rates are often quoted. Data transfer rate is quoted in Kilobytes/second, Megabytes/second, or Gigabytes/second.
- *Zone bit recording*, a technique which enables data to be recorded at variable densities throughout the disk, thereby providing higher capacities. This method breaks the surface of the disk into different zones, each with a different number of bytes per track, enabling the bit density to remain constant across the disk's surface. Drives with this feature offer improved transfer rate and capacity.
- *Bus bandwidth*, the amount of data that can be handled by the system bus. SCSI bus bandwidth can be made faster and wider.
- *Controller*, a device which coordinates read and write requests between the host and the disk drive.
- *Buffer size*, the amount of cache employed on the disk controller. Some drives utilize separate read and write buffers; others use a unified cache.
- *Mean Time Between Failure (MTBF)*, the average time between failures.
- *Mean Time to Data Loss (MTDL)*, the average time a disk subsystem can operate before another failure causes data to be lost.

More information on these and related topics can be found in *Sun Performance and Tuning, Java and the Internet, Chapter 8: Disks*, and *Configuration and Capacity Planning for Solaris Servers, Chapter 7: Storage Subsystem Architecture*.

Disk Products from Sun

Sun offers a wide variety of storage products, ranging from single disk drives to large array subsystems. Descriptions of the mass storage products available currently from Sun can be found at *http://www.sun.com/storage/products.html* on the World Wide Web.

Disk Interface Standards

One critical factor in the ability of a disk subsystem to perform lies in its connection to the host system. A number of disk interface standards are in use today, including EIDE, SCSI, and Fibre Channel.

EIDE

The Enhanced Integrated Drive Electronics (EIDE) peripheral interface has gained popularity with low-end systems. Running at 16.7 MB/second, EIDE offers improved performance at significantly lower cost than comparable SCSI disk drives. However, EIDE does not scale as well as SCSI as fewer devices can be connected to a bus.

EIDE is offered only in Sun's Ultra 5 and Ultra 10 systems.

More information on EIDE as implemented on Sun systems can be found in the *Ultra 5 and Ultra 10 Workstation Architecture* white paper located at http://www.sun.com on the WorldWide Web.

Small Computer Systems Interface (SCSI)

The Small Computer Systems Interface (SCSI) has emerged as the preeminent peripheral interface over the last decade. Initially designed to support lower cost systems, SCSI has evolved, providing sufficient bandwidth for even the most demanding server systems.

The most widely used SCSI interfaces include the original 8-bit, 5 MB/second SCSI-1 bus protocol, an enhanced version called SCSI-2, the 16 MB/second Fast SCSI interface, 16-bit Fast and Wide SCSI, and Differential SCSI. UltraSCSI, the latest version of SCSI to date, is a 16-bit interface that provides 40 MB/second transfer rates to applications.

Table 4-1 highlights the key characteristics of the various SCSI standards.

	SCSI	Fast SCSI	Fast and Wide SCSI	Ultra SCSI
Asynchronous/ Synchronous	• Asynchronous or Synchronous • Single-ended	• Synchronous • Differential	• Synchronous • Differential	• Synchronous • Differential
Transfer Rate	• 5 MB/second	• 10 MB/second	• 20 MB/second	• 40 MB/second
Number of Pins	• 50 pins	• 50 pins	• 68 pins	• 68 pins
Width	• 8 bits	• 8 bits	• 16 bits	• 16 bits
Maximum Cable Length	• 6 m	• 3 m synchronous • 25 m differential	• 3 m synchronous • 25 m differential	• 3 m synchronous • 12 m differential
Number of Targets	• 7	• 7	• 15	• 15

Table 4-1 The SCSI protocols provide a wide variety of performance characteristics

More information on SCSI can be found in *Configuration and Capacity Planning for Solaris Servers, Chapter 7: Storage Subsystem Architecture*.

Fibre Channel

Fibre Channel is a high performance interconnect standard designed for bidirectional, point-to-point communications between high performance workstations, peripherals, and large host systems. It offers a variety of benefits over other link-level protocols:

- Efficiency and high performance
- Scalability
- Operation over long distances
- Easy to cable
- Supports popular high level protocols
- Supports switched networks
- Secure from RF snooping
- Invulnerable to RF interference

Theory of Operation

Channels are a generic class of communications technology designed to move data from one point to another with high speed and minimum latency. For speed, functions such as error correction and retry-on-busy are typically implemented in hardware. Because they do not have sophisticated software-based routing and switching capabilities, channels typically operate in environments where every device on the channel is defined and known in advance.

At its most basic, Fibre Channel is just a high speed serial connection between two buffers. Indifferent to the format of the data, Fibre Channel only provides the error checking and necessary signaling to ensure the reliable movement of data between the source and destination buffers.

Fibre Channel employs a mechanism known as a *fabric* to establish connections between *ports*. The fabric, like the telephone network, has all the needed intelligence to make routing decisions. The only concern of a port is to ensure its proper connection to the fabric. Each fabric has an address space that allows up to 16 million addresses. A fabric can be simple, or a mix of switches and hubs that connect a complex and widely dispersed network of ports.

Reliability

There are no limits on the size of a transfer between applications using Fibre Channel. Each sequence is broken into frames, the framesize being negotiated by the ports and the fabric. A 32-bit cyclic redundancy check ensures that transmission errors can be reliably detected.

Fibre Channel Arbitrated Loop

An important enhancement to Fibre Channel has been the development of Fibre Channel Arbitrated Loop (FC-AL), developed specifically to meet the needs of storage interconnects. Employing a simple loop topology, FC-AL can support both simple configurations and sophisticated arrangements of hubs, switches, servers, and storage systems (Figure 4-1). Furthermore, by using SCSI protocols over the much faster, more robust Fibre Channel link, FC-AL provides higher levels of performance without requiring expensive and complex changes to existing device drivers.

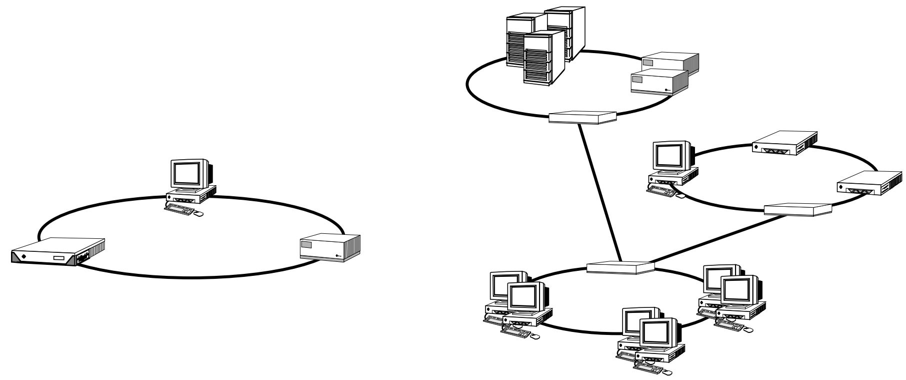

Figure 4-1 FC-AL's loop topology can support both simple and complex configurations.

Key Benefits of FC-AL

FC-AL offers a host of benefits over other protocols, enabling storage systems based on it to foster the creation of new applications that take full advantage of some impressive specifications:

- *Gigabit bandwidth*

  FC-AL is capable of data transfer rates of up to 100 MB/second (dual ported), with 400 MB/second envisioned for the future — far faster than SCSI, Serial Storage Architecture (SSA), or P1394 (Firewire).

- *Suitability for networks*

  In addition to performance, Fibre Channel is ideal for building networks. Employing hubs and switches similar to those used in networks, Fibre Channel enables complex arrangements of storage and systems to be connected together in highly scalable, highly available networks, or fabrics.

- *Use of existing SCSI protocols*

  FC-AL allows SCSI command packets to be sent over a high speed physical medium, reducing software and firmware costs and minimizing impact on existing software.

- *Node addressability far better than SCSI*

  With the ability to support up to 126 FC-AL devices on a single host adaptor, cost and implementation complexity is greatly reduced. Using optical fiber media, a single FC-AL loop can support nodes with a separation of up to ten kilometers from the host.

- *Unique device naming convention*

  FC-AL employs unique, 64-bit world wide names for devices. Disk drives, host adapters, and interface boards are identified by a world wide name, enabling improved diagnostics and software support.

- *Greatly simplified wiring and cabling requirements*

  Because Fibre Channel is a largely optical, simple serial protocol, electrical interference and expensive cabling are much less of an issue than with the complex parallel data paths used by SCSI.

In addition, FC-AL implementations support redundant data paths, hot pluggable components, multiple host connections, and dual ported drives — features that 15 year-old SCSI technology was never intended to support.

Configuration Flexibility

Fibre Channel Arbitrated Loop enables storage systems to implement several loop configurations, each designed to address varying levels of performance and availability. Utilizing bypass circuits and multiplexors, FC-AL based storage devices like the Sun StorEdge A5000 permit the construction of multiple loops within a single enclosure, improving both performance and availability (Figure 4-2).

More information on Fibre Channel technology can be found in *Sun Performance and Tuning, Java and the Internet, Chapter 8: Disks*, and *Configuration and Capacity Planning for Solaris Servers, Chapter 7: Storage Subsystem Architecture*, and in the white papers listed in the References section of this document.

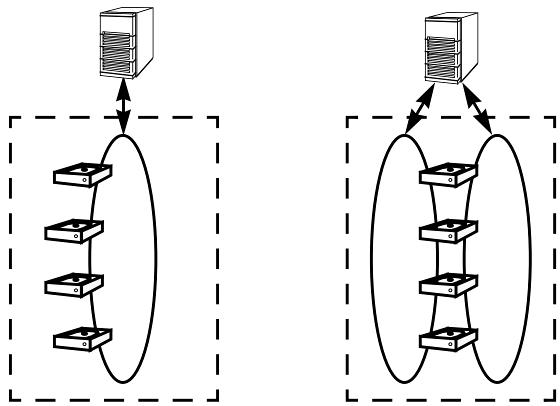

Figure 4-2 FC-AL's loop topology supports both single and dual loop configurations.

RAID

Redundant Arrays of Inexpensive Disks, or RAID systems, were first formally defined by researchers at Berkeley in 1987. The most important part of their idea was to combine the power of small, inexpensive drives to achieve performance equal to a single large, expensive disk. Because RAID required multiple spindles, features were also needed that could protect against the data loss that could be caused by the failure of any one of the drives. These original safeguards subsequently became an important feature in their own right, with many users purchasing RAID systems just for their availability.

Today, disk arrays are sold as an intelligently managed collection of disk drives organized to optimize performance for a particular set of tasks. All RAID systems achieve higher performance and/or reliability by replicating or spanning data across multiple disks. Exactly how this is done has profound effects on subsystem performance, cost, and reliability.

Two metrics are used in assessing the performance of disk subsystems:

- *Transfer Rate*

  Transfer rate is the speed (measured in MB/second) with which a subsystem can move data through its controller. In RAID systems, read and write transfer rate performance can vary considerably, and is particularly valuable for applications that have to move large amounts of data quickly, like document imaging or digital video.

- *I/O Operations per Second (IOPS)*

  IOPS are a measure of the ability of a storage system to handle multiple, independent I/O requests in a certain period of time. RAID systems with high transfer rates do not always have good transaction rate performance. Database and transaction processing systems are examples of applications that typically require high I/O rate performance.

Recovery After a Failure

When a disk in a protected RAID system does fail, the array continues to operate in *reduced mode*. In reduced mode, array performance may be lowered, depending on the RAID level being used and whether reads or writes predominate. Upon encountering a failure, the simplest RAID systems inform the administrator of the problem, leaving them to organize the replacement of failed components and to begin the restoration of normal operations. The most sophisticated systems can automatically begin reconstructing data to a reserve drive called a *hot spare*. Hot sparing allows the array to automatically return to full performance quickly, and permits administrators to defer maintenance to a more convenient time. Some systems even allow defective drives to be replaced without interrupting processing or removing power, but such *hot swapping* is sometimes less useful than a system that can instantly begin recovery using a hot spare.

Popular RAID Levels

Can RAID subsystems actually eliminate the I/O bottleneck and provide very high levels of performance? In some circumstances, RAID configurations can dramatically improve performance and in others may actually impair it. To accommodate the various kinds of demands put on storage systems, different *levels* of RAID have been defined, each with their own special characteristics.

Five RAID levels, numbered 1 through 5, were defined by the Berkeley researchers. Since that time, a few more have been added — some simply variations of the original five. The four RAID levels offered in the Sun StorEdge array product family, levels 0, 1, 5, and 0+1, are discussed here.

Level 0 - Striping

Striping is an industry term for breaking a data stream up and placing it across multiple disks in equal-sized *chunks* or *stripe blocks*. Each chunk is sequentially written to successive drives in the array. The set of sequential chunks that begins with the first drive and ends with the last drive forms the “stripe” (Figure 4-3). Array management software running on the host, on a special controller inside the array, or some combination of both, is responsible for making the array look like a normal, or *virtual*, disk to the operating system and applications.

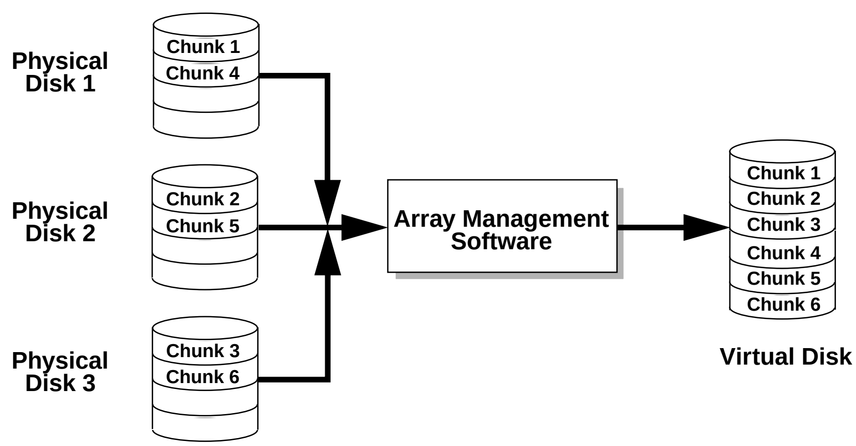

Figure 4-3 RAID 0 breaks data into equal sized “chunks” of data in sets that form a “stripe”. In this figure, the stripe width is three, with chunks 1, 2, and 3 forming a single stripe.

Each disk in a stripe is generally assumed to be on its own independent data channel, allowing the transfer rate of a RAID 0 implementation to approach the sum of the transfer rates of each of the drives. (Presuming that the intervening software and hardware has sufficient throughput.)

RAID 0 can also provide excellent IOPS performance, because data is spread across many independent spindles and actuators, helping to balance the I/O load. Optimizing RAID 0 for transfer rate or IOPS performance requires administrators to make configuration decisions about the relationship of the sizes of the chunks in a stripe and the average size of disk transactions.

Level 1— Mirroring

Used for a long time in fault tolerant applications, this type of array is also called *shadowing* or *mirroring*. Designed to optimize data availability rather than speed, RAID 1 duplicates each write transaction to one or more “mirror” disks (Figure 4-4).

Mirrored systems can be used to improve the IOPS performance of read operations, because the least busy drive can be selected to service requests. Because both drives must be involved in write operations, write performance may be slightly worse than an independent drive.

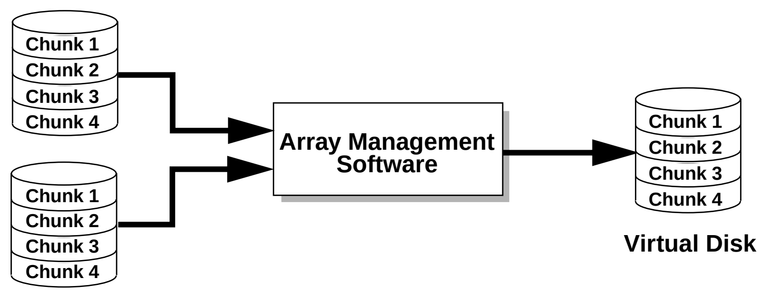

Figure 4-4 RAID 1, or mirroring, provides data redundancy by recording data multiple times on independent spindles.

Level 5 —Striping with Interleaved Parity

RAID 5 exploits the performance gains of striping while mitigating its vulnerability to data loss by adding error correction information (called *parity*) to the data.

RAID 5 allows independent access to individual drives in a group with striped data. If the data is interleaved across the group of drives in blocks equal in size or larger than most I/O requests, then drives can respond independently to requests, and even to multiple requests simultaneously. This has the effect of allowing RAID 5 to have very good random read performance. Conversely, if the block sizes in a stripe are smaller than the average size of an I/O request, forcing multiple drives to work together, RAID 5 systems can also enjoy good sequential performance.

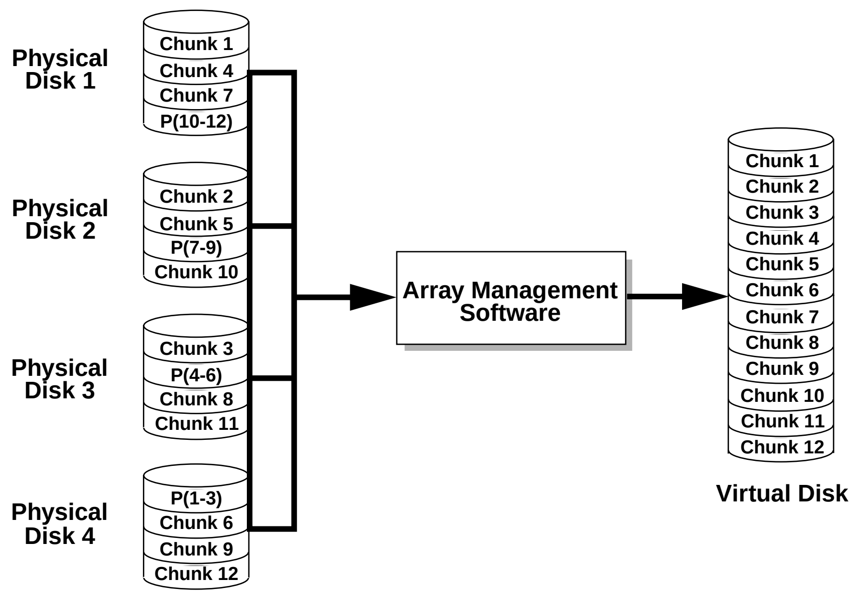

Figure 4-5 RAID 5 interleaves parity and data across the stripe to improve performance.

With RAID 5, both the parity information and data are interleaved across the array in a cyclic pattern (Figure 4-5). If one of the disks fail, a simple algorithm allows the missing data to be recovered from the remaining disks. RAID 5 is a good compromise for those needing good random performance without incurring the expense of mirroring.

RAID 5's failing comes from its poor performance on small writes. When a single-block write command is issued to a RAID 5 configuration, the system must first complete four I/O operations and two parity calculations in a *read-modify-write* sequence, degrading random write performance.

Cached RAID 5

Using a non-volatile cache in the disk controller, the need to modify write operations can be optimized to use full width stripes, resulting in very good write performance. This is often known as hardware RAID 5 or controller-based RAID. Sun's StorEdge A1000, A3000, and A7000 subsystems perform high speed cached RAID 5 operations for write-intensive workloads.

RAID 0+1 — Striping plus Mirroring

Useful configurations have been created by combining two existing RAID levels. One of the more popular is known as RAID 0+1, which combines the reliability of mirroring (RAID 1) with the performance of striping (RAID 0) (Figure 4-6).

The reliability of a striped and mirrored system is excellent because of the high redundancy afforded by mirroring. RAID 0+1 systems can tolerate the failure of one or more disks, and continue to deliver data with virtually no performance degradation, unlike RAID 5. RAID 0+1 systems do carry, however, the higher cost of mirrored systems, as data requiring protection needs twice the disk space of simple independent spindles.

Some subsystems also support RAID 1+0, in which data is first mirroed and then striped. This scheme can withstand more individual disk failures and mantain data availability.

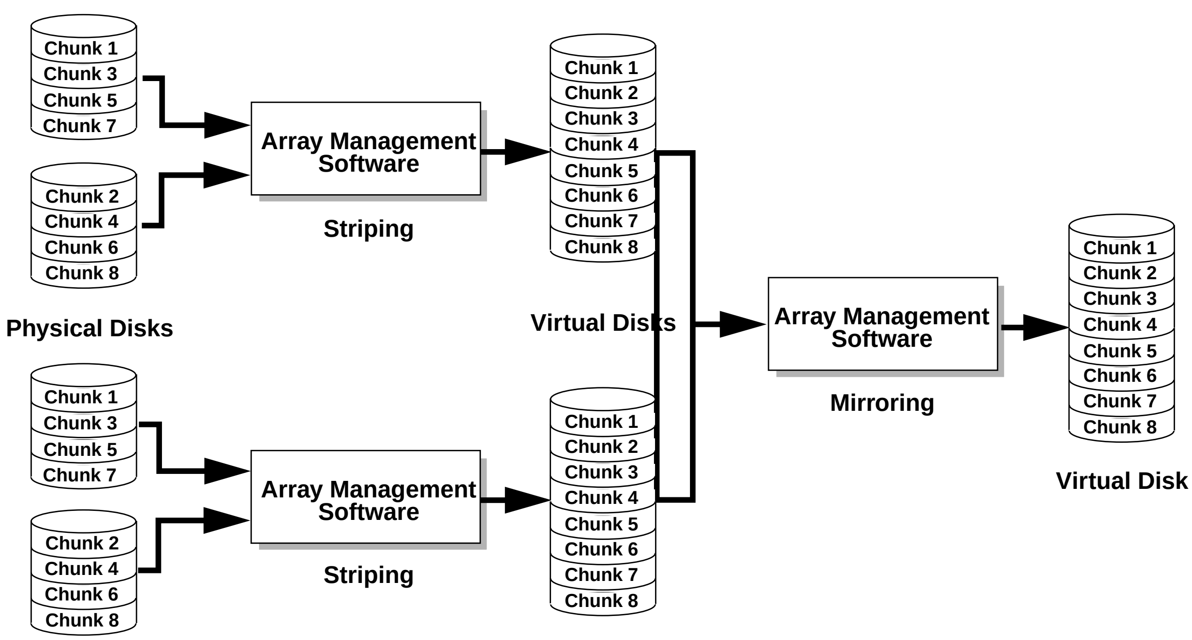

Figure 4-6 RAID 0+1 combines mirroring and striping to achieve excellent availability and performance.

Optimizing the RAID Configuration

The various RAID levels have distinctive strengths, and therefore tend to be best for certain kinds of applications (Table 4-2). Database management, for example, requires good random performance, and independent spindles are the best solution. If reliability is a concern, however, mirroring (RAID 1) can give excellent performance, but is expensive for large data sets. RAID 5 offers slightly worse throughput, but with good redundancy and economy. Clearly, finding the right solution requires a good understanding of the I/O load.

	Strengths	Weaknesses
RAID 0 **Striping**	• Improves I/O performance	• No redundancy • A single disk failure causes all data to be lost
RAID 1 **Mirroring**	• Improves data availability • May improve read performance	• Expensive • Potentially decreased write performance
RAID 0+1 **Striping Plus MIrroring**	• Improves data availability without sacrificing performance	• Expensive
RAID 5 **Striping Plus Distributed Parity**	• Improves data availability • Error correction information • Lower cost than mirroring	• Survives only a single disk failure • Poor small write performance

Table 4-2 RAID levels provide varying degrees of availability with corresponding trade-offs in performance and cost

Interlace Size

The interlace size specifies the *chunk size*—the amount of data written to each disk before moving on to the next component in a stripe. Optimally setting the interlace size can help ensure the RAID configuration performs well for both read and write operations. RAID 0 and RAID 5configurations require different settings for the interlace size, depending on the type of I/O:

- In striped configurations experiencing a high volume of random access I/O, the interlace size should be set large relative to the size of a typical I/O request.
- In striped configurations subject to mostly sequential access I/O, the interlace size should be set small relative to the size of the typical I/O request. One typical strategy to determine the interlace size is to divide the typical I/O size by the number of disks in the stripe (Figure 4-7).

$$\text{Interlace\_Size} = \frac{\text{I/O\_Size}}{\text{Number\_of\_components}}$$

Figure 4-7 Setting the interlace size in striped configurations fosters higher sequential performance

- RAID 5 configurations perform best when write requests align with the stripe and the I/O is a multiple of the interlace size. Indeed, full stripe writes can be utilized if the size of a typical I/O request equals the interlace size multiplied by one less than the number of components in the stripe (Figure 4-8).

$$\text{I/O\_Size} = \text{Interlace\_Size} * (\text{Number\_of\_components} - 1)$$

Figure 4-8 Setting the interlace size in RAID 5 configurations promotes faster write operations

More information on configuring and optimizing RAID systems can be found in *Sun Performance and Tuning, Java and the Internet, Chapter 8: Disks*, and *Configuration and Capacity Planning for Solaris Servers, Chapter 7: Storage Subsystem Architecture*.

Host-Based RAID versus Controller-Based RAID

In recent years, much has been said about the implementation options available for RAID subsystems, and contention has arisen over the perceived benefits and drawbacks to both host-based RAID and controller-based RAID devices. While traditional controller-based RAID subsystems employing SCSI technology were long seen as the answer to higher performance by minimizing the amount of traffic flowing between the host and peripheral, the advent of 100 MB/second Fibre Channel is bridging the gap. Utilizing a faster transport medium like Fibre Channel, the performance of host-based RAID storage systems compete with, and in many cases surpass, their controller-based RAID counterparts. In addition, the lower cost and higher degree of flexibility offered with host-based RAID solutions like the Sun StorEdge A5000 ensures that users have access to the best price/performance configurations available.

Disk Activity and Performance — iostat

Two factors that impact overall system throughput are CPU utilization and disk I/O activity. If there are insufficient CPU resources to process data and execute tasks, data will not be streamed to devices and performance will suffer. In addition, it is possible for disks to become saturated, rendering them incapable of handling higher loads of traffic.

The `iostat` utility bundled in the Solaris operating environment helps system administrators determine disk activity patterns and performance. In order to iteratively report disk I/O activity as well as CPU utilization, the Solaris kernel maintains a series of counters:

- The number of reads and writes per disk
- The number of bytes read and bytes written per disk
- Time stamps
- The residence time and cumulative residence length product for each queue

These statistics enable `iostat` to produce highly accurate measures of throughput, utilization, queue lengths, transaction rates, and service times (Figure 4-9).

```
# iostat -xp
                                                  extended device statistics
device    r/s   w/s   kr/s   kw/s   wait   actv   svc_t   %w    %b
dad0      0.0   0.0   0.0    0.0    0.0    0.0     0.0    0     0
dad0,a    0.0   0.0   0.0    0.0    0.0    0.0     0.0    0     0
dad0,b    0.0   0.0   0.0    0.0    0.0    0.0     0.0    0     0
dad0,c    0.0   0.0   0.0    0.0    0.0    0.0     0.0    0     0
dad1      0.0   0.2   0.1    1.4    0.0    0.0    61.7    0     0
dad1,a    0.0   0.2   0.1    1.4    0.0    0.0    61.7    0     0
dad1,b    0.0   0.0   0.0    0.0    0.0    0.0     7.9    0     0
dad1,c    0.0   0.0   0.0    0.0    0.0    0.0     0.0    0     0
fd0       0.0   0.0   0.0    0.0    0.0    0.0     0.0    0     0
sd2       0.0   0.0   0.0    0.0    0.0    0.0     0.0    0     0
nfs1      0.0   0.0   0.0    0.0    0.0    0.0     0.0    0     0
```

Figure 4-9 The `iostat` utility reports I/O statistics which can be analyzed to determine CPU and disk bottlenecks.

More information on `iostat` can be found in the `iostat(1M)` man page.

Managing Disk Space

Data availability depends on disk drives to be able to read and write data at a given time. System administrators must ensure that adequate disk and file system space exists for users and applications. Quotas can be used to constrain users from utilizing too much disk space.

Determining Free Disk Blocks and Files — df

To determine the number of free disk blocks and files on the disks in the system, administrators can employ the `df` command included in the Solaris operating environment. The `df` command displays the amount of disk space occupied by currently mounted file systems, the amount of used and available space, as well as the amount of each file system's capacity that has been used (Figure 4-10).

```
# df -ak

Filesystem              kbytes    used      avail   capacity  Mounted on
/proc                   0         0         0       0%        /proc
/dev/dsk/c0t1d0s0       3875742   816409    3020576 22%       /
fd                      0         0         0       0%        /dev/fd
swap                    637832    55312     582520  9%        /tmp
auto.shared             0         0         0       0%        /shared
auto.scde               0         0         0       0%        /scde
auto.src                0         0         0       0%        /src
-hosts                  0         0         0       0%        /net
auto.doe                0         0         0       0%        /doe
auto.ws                 0         0         0       0%        ws
auto_home               0         0         0       0%        /home
-xfn                    0         0         0       0%        /xfn
nevermore:vold(pid236)  0         0         0       0%        /vol
```

Figure 4-10 The `df` utility reports free disk blocks and files, enabling administrators to quickly isolate capacity problems that can lead to decreased performance and availability.

More information on `df` can be found in the `df(1M)` man page.

Determining Disk Usage — du

The ability to ensure that disks do not completely fill aids both performance and availability. System administrators can obtain a quick gauge of disk usage by executing the `du` command. The `du` command recursively describes the number of kilobytes contained in all files and directories (Figure 4-11). The size of the file space allocated to a directory is reported as the sum of the total space allocated to the files contained in the directory, as well as the space allocated to the directory itself. All files sizes are reported in, and rounded to, the nearest 512 byte unit.

```
# du -as /tmp
306128     /tmp
```

Figure 4-11 The `du` utility describes the amount of disk space allocated to files and directories.

More information on the `du` command can be found in the `du(1M)` man page.

System Activity Reporter — sar

The Solaris operating environment includes a system activity reporter (`sar`) that samples cumulative activity counters at specified intervals. The `sar` command can also be used to determine storage related information, including:

- Disk activity levels for system disks (`-d` option)
- Buffer cache activity (-b option)
- File access system routine usage (`-a` option)
- Process status (`-v` option)
- Average queue length (`-q` option)

More information on the `sar` command can be found in the `sar(1M)` man page.

Volume Management

The combination of disk drives and intelligent software to manage them is commonly referred to as a *storage management system*. With increasing numbers of disks and a higher demand for availability in mission-critical environments, storage management systems have a number of requirements:

- *Reliability*

  Organizations need assurance that the data they depend on is accurate. Storage management systems must implement the features necessary to ensure the accurate storage and retrieval of data.

- *Data and system availability*

  Many organizations run 7x24. In these environments, it is imperative for data to be available on demand. For many, unavailable data can mean a significant loss in income. When a system fails, data must be accessible to an alternate host. In the event of a disk failure, some means must exist for the data stored on that disk to be made available to users.

- *Performance*

  Storage management systems must provide the performance required by today's demanding applications and environments.

- *Management of large amounts of data*

  Users are generating more data than ever before. Organizations need a method for organizing disks and the data on them such that data reliability and availability are maintained without sacrificing performance.

- *Simplified administration*

  Managing a large number of disks can be daunting and challenging. Storage management systems must provide tools to ease this task.

- *Application integration*

  A storage management system must fit seamlessly into existing customer server, database and OLTP environments.

Sun provides two tools to aid these efforts: the Sun StorEdge Volume Manager, and RAID Manager. More information on Sun volume management solutions can be found at *http://www.sun.com/storage/products.html* on the World Wide Web.

File System Tuning 5

The Solaris operating environment supports several file system types, each designed to provide high performance to users. These file systems include the standard Unix File System (UFS), a RAM disk file system (Tmpfs), and the VERITAS file system (VxFS) discussed below. Two additional file systems, CacheFS and NFS, are geared toward enhancing the performance of network operations and are discussed in chapter 7 of this document.

Unix File System

The Unix File System (UFS) is the standard file system used to manage disk drives in the Solaris operating environment. Based on the Berkeley Fast File System, UFS is designed for high performance. Its tree-like structure, managed by a structure known as an *inode*, enables fast access to directories and files. With UFS, all files that are read are stored in RAM. If RAM becomes full, files are flushed to disk and retrieved when needed. Consequently, the more RAM in the system the better overall file system performance.

UFS and Transaction Logging

When unscheduled system downtime occurs, such as a system panic or power failure, partially completed updates to the file system are possible. If partial updates exist, the file system can be left in an inconsistent state. To fix these inconsistencies, UFS file systems are checked at boot time (with `fsck` and `quotacheck`) and repaired, if possible. Because multiple and complex directory hierarchies often exist, and because `fsck` must traverse each and every hierarchy to ensure its integrity and consistency, the `fsck` process is often time consuming.

The combination of Solaris 2.4 and the Solstice DiskSuite storage management tool offers a new feature called UFS Logging, sometimes referred to as journalling. UFS logging takes the (logically) synchronous nature of updating a file system and makes it asynchronous. Updates to a file system are not made directly to the disk or partition containing the file system. Instead, user data is written to the device containing the file system, but the file system disk structures are not modified—they are logged instead.

Updates are applied to the file system structure at key times to ensure consistency. Changes made to the file system by unfinished system calls are discarded, ensuring that the file system is in a consistent state. This means that logging file systems do not have to be checked at boot time, speeding the reboot process.

System administrators should consider the following when configuring UFS file systems with transaction logging:

- The absolute minimum size for a transaction log is 1 MB of disk space. A recommended minimum is 1 MB of log space for every 1 GB of file system space.
- The larger the log, the more simultaneous file system operations per second. As a general rule, administrators should plan on 1 MB of log space for every 100 MB of file system space being logged.
- Small file systems, or primarily read-intensive file systems, will not benefit from transaction logging. However, file systems experiencing a significant number of updates should be logged.

More information about UFS transaction logging can be found in the *Solstice DiskSuite User's Guide*, and *Solstice DiskSuite Reference Guide* available at *http://www.docs.sun.com* on the World Wide Web.

Configuring UFS File Systems for Maximum Performance

The advent of very large disk drives and RAID subsystems is leading to ever increasing file system sizes. Users of large file systems need to reset a number of UFS parameters to ensure efficient disk utilization and I/O performance:

- *inode density*

  When UFS was introduced, most files were very small, averaging only 1 to 2 KB in size. Today, files are approaching 64 KB or larger. To ensure the file system utilizes space efficiently, system administrators should set the inode density to one inode per 8 KB of file space.

- *minfree*

  Disk have grown tremendously since the `minfree=10%` rule was set back in 1980. At that time, 10 percent of a disk ranged from 2.5 MB to 5 MB. Today, 10 percent of a typical disk is several hundred megabytes. For large file systems, or those using disk drives larger than 2 GB, `minfree` should be set to 1 percent to reduce file system overhead and wasted disk space. On large file systems, `minfree` is automatically reduced (Solaris 2.6 and later).

- *maxcontig*

  The file system cluster size, or `maxcontig`, specifies the maximum number of blocks belonging to a single file that can be allocated contiguously before inserting a rotational delay of the disk. Performance may be improved if the file system I/O cluster size is an integral of the stripe width of file systems on RAID 0 or RAID 5 volumes.

- *cylinders per cylinder group*

  Volumes and file systems that utilize more than 8 GB of disk space may need to increase the size of a cylinder group from 16 cylinders per cylinder group to as many as 256 cylinders per cylinder group.

More information on configuring UFS file systems can be found in...?

Tmpfs

The Solaris operating environment include support for Tmpfs, a temporary file system that utilizes memory to cache writes scheduled to go to disk. System administrators can improve system performance by making extensive use of Tmpfs to store short-lived files. This scheme enables temporary files to be written and accessed without the typical performance impact associated with writing information to disk.

A system is a candidate for Tmpfs usage if:

- If it runs applications that create temporary, short-lived files. (Systems that run compilers are a good example of this type of environment.)
- It has a lot of memory and swap space.
- It employs slow disks.

Since many applications already employ the `/tmp` directory on Solaris systems for temporary files, it is a natural candidate for Tmpfs. System administrators should note, however, that any directory can utilize Tmpfs.

More information on Tmpfs can be found in...?

VERITAS File System

The VERITAS File System is a high-performance, fast recovery, enterprise file system. Typically employed with very large mass storage subsystems from Sun, the VERITAS File System offers a host of features and benefits to users, including:

- *Enterprise file system*

  The VERITAS File System is designed as an enterprise file system, providing superior file system performance in certain applications and increased availability over the standard UNIX file system (UFS).

- *Availability*

  High availability is an essential requirement of enterprises running mission-critical applications. The VERITAS File System provides fast recovery after system crashes, decreasing system down-time and increasing data and application availability. Administrative operations such as file system backup, resizing, and defragmentation can occur on-line. Administrators no longer need to take the file system off-line to perform these common operations, increasing file system availability and making administration more convenient.

- *High performance*

  Using a more efficient extent-based allocation scheme, hashed directories, as well as layout and caching modifications, the VERITAS File System provides the high I/O performance demanded by large enterprise databases and other applications.

Extent-Based Allocation

The VERITAS File System allocates space in contiguous segments, or *extents*, instead of small fixed-size disk blocks. This can accelerate sequential I/O by reducing seek and disk rotation times. Performance is also improved as fewer I/O operations are needed to read and write large amounts of data.

Extent sizes can be chosen automatically by the VERITAS File System based on the I/O pattern of the file, or can be explicitly selected by the administrator.

More information on the VERITAS File System can be found at *http://www.sun.com/storage/software/veritas.html* on the World Wide Web.

Tuning File Systems—tunefs

The layout of a file system can have a dramatic affect on I/O performance. When a file system is over 90 percent full, system administrators should optimize it for time, ensuring higher throughput. The `tunefs` utility is designed to change the dynamic parameters of a file system than affect layout policies.

More information on `tunefs` can be found in the `tunefs(1M)` man page.

Networking 6

The last decade has seen a ten-thousand fold increase in computer performance. This increase in power, along with advances in operating systems and graphical desktop environments, has spurred the development of an array of new software tools and technologies. These new capabilities in software have in turn pushed computing resources to their limit, causing users to demand higher performance hardware. Networks are developing through a similar technology and demand-driven cycle.

Networks must not only keep pace with technological advancements, they must ensure that sufficient resources are available to meet the demands of increasingly sophisticated applications. To meet these needs, networks must perform well and respond to client requests as needs dictate.

When a network server is unable to respond to a client's request, the client typically retransmits the request a number of times. Each retransmission induces additional system overhead and generates more network traffic. System administrators can mitigate the extent of excessive retransmissions by improving three key conditions: data integrity, system performance, and network congestion.

Selecting the Right Network Technology

Local Area Networks

Local Area Networks, or LANs, provide a means to connect systems within a limited distance, facilitating information and resource sharing in a multivendor environment. While adequate for small, centralized environments, existing LAN technologies are quickly becoming a bottleneck in larger distributed computing environments.

In the future, with resources as likely to be remote as local, networks must be able to accommodate the rapid movement of large amounts of data. With escalating needs for collaboration, organizations are demanding advanced functionality, such as video multicasting, time-synchronous applications, and integration with wide area networks. Enterprises require high speed networks (100 Megabits/second or above) but will not accept expensive changes to their existing hardware, software and training. To keep up with the expanding demands of computing, Sun believes that three technologies have promise: Ethernet, FDDI, and ATM.

Ethernet

Conventional Ethernet was designed when it was ambitious for an affordable departmental system to deliver 1 SPECmark. Provided as a standard feature or low-cost option with nearly every workstation, 10 Megabit/second Ethernet (Ethernet or Ethernet 10) is a ubiquitous, inexpensive alternative which is simple to deploy. Its popularity has made possible the economies of scale that come from volume manufacturing, which in turn has made Ethernet inexpensive.

Although it has extensive capabilities, such as ease of connectivity, and a stable installed base, Ethernet is inadequate for today's ambitious requirements (Figure 7-2). Its inability to scale to more than a few hundred users, or easily accommodate bandwidth-intensive applications makes it difficult to use in many environments.

Because Ethernet is collision-based, only one device can transmit at a given time. Should two devices attempt to transmit simultaneously, their data packets will collide, resulting in lower aggregate throughput. A 10BaseT Ethernet network is considered saturated at 35 percent utilization.

Consequently, approximately 500 KB of traffic can saturate a 10BaseT LAN. The use of switches and routers can minimize collisions and improve LAN performance by breaking the network into pieces and isolating traffic to smaller segments.

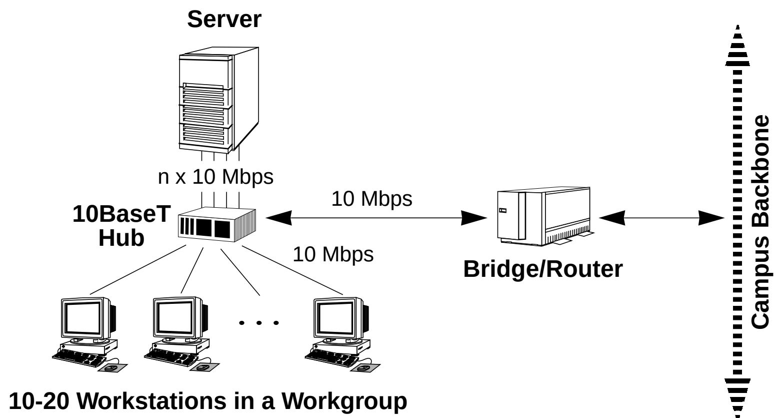

Figure 7-1 Today, networking technology is characterized by inadequate performance, complex interconnections, and multiple protocols

Fiber Data Distributed Interface

Primarily used to satisfy the needs of data communication users, the Fiber Data Distributed Interface (FDDI) has become a standard for networks requiring 100 Megabit/second transmission speeds. Providing media access control, data encoding and decoding, as well as defining drivers for fiber-optic components and a multilayered network management scheme, FDDI often is used as the backbone for lower speed networks such as Ethernet 10.

Although a good investment when speed is required, FDDI has limitations. It is more expensive to deploy than Ethernet, it offers limited bandwidth, and is not as popular, and therefore is more expensive than other networking technologies. Nevertheless, for users that require standardization, a high degree of interoperability, flexibility and robustness, FDDI is one of the most mature high speed network technologies available today.

Fast Ethernet

For users needing a lower-cost alternative to FDDI, Fast Ethernet (100 Megabit/second Ethernet) may be the networking technology of choice. Fast Ethernet is an extension to the Ethernet 10 standard, supporting a wide range of applications with increased bandwidth. Fast Ethernet maintains compatibility with the installed base of wiring currently employed for Ethernet. It has the ability to automatically sense 10 Megabits/second or 100 Megabits/second operation and can adjust to either speed. Fast Ethernet ensures maximum compatibility with existing environments and applications, providing an investment-preserving migration path from conventional Ethernet.

Gigabit Ethernet

The transition of faster interfaces, such as Fast Ethernet, to a widespread desktop technology had created another problem for network administrators. As the number of devices entering the network at 100 Mbps grows, the demand for higher speed at the backbone and server level also increases. Organizations are finding they need to increase the level of network performance to support computing power demands.

Expected to become the solution of choice for network administrators, Gigabit Ethernet provides a raw data bandwidth of 1000 Mbps (or 1 Gigabit per second) that aims to help the increased network congestion experienced at the backbone and server levels. Not just a high performance networking technology, Gigabit Ethernet also gives administrators the flexibility they need — new full-duplex operating modes for switch-to-switch and switch-to-server connections, and half-duplex modes for shared connections using repeaters and the carrier sense multiple/collision detection (CSMA/CD) access method.

More information on Gigabit Ethernet and its application can be found in the *Sun and Gigabit Ethernet* white paper located on the World Wide Web at *http://www.sun.com/products-n-solutions/hw/networking/whitepapers/gigabitwp.html*.

Asynchronous Transfer Mode

Perhaps the most promising new network technology is Asynchronous Transfer Mode (ATM). Based on cell switching technology, ATM uses small, fixed-sized cells and inexpensive, high-bandwidth, low-latency switches. With data rates from 51 Megabits per second to over 2 Gigabits per second, ATM offers the performance needed by the most demanding interactive applications (Figure 7-2).

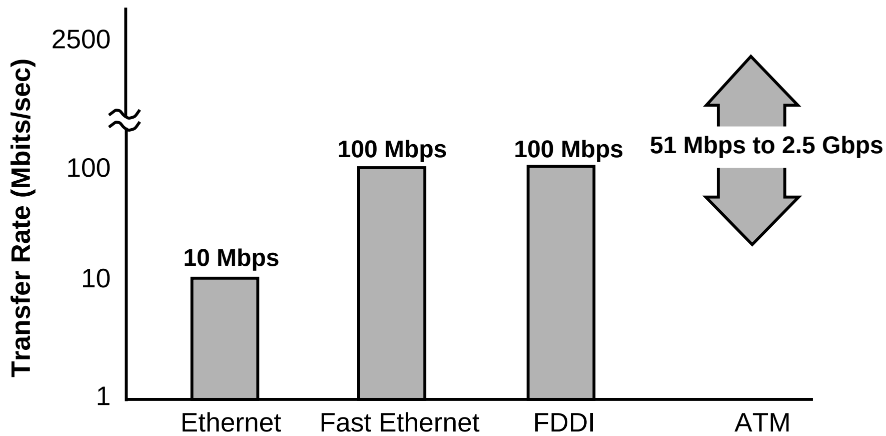

Figure 7-2 Advanced networking technologies have performance far exceeding that of Ethernet, the most commonly used network today

ATM is independent of the speed of the line on which it runs, making it adaptable to most wiring technologies. ATM can run full duplex, permitting the simultaneous transmission and receipt of data without interruption. Its flexibility and low latency allows real-time applications like video and audio to be supported along with data services.

A scalable technology offering a deterministic quality of service, ATM allows the network to be tailored to the application. A global, mobile, scalable, and application transparent protocol, ATM allows the mixture of data, voice and video. With its increased performance and ability to support advanced applications, ATM holds the promise of being the premier network solution for the rest of the decade and beyond (Figure 7-4).

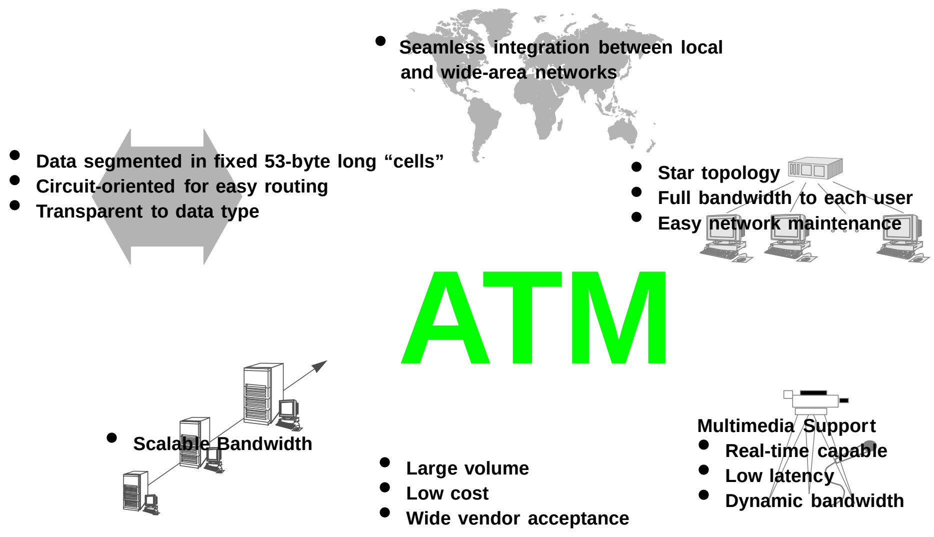

Figure 7-3 ATM has many characteristics that make it the preferred solution for most future networking requirements

Wide Area Networks

Wide Area Networks, or WANs, provide a means to connect systems which are distributed over a large geographic area, facilitating information and resource sharing. WANs must be low-cost, high speed, and easily integrated with LANs. Three technologies are emerging as leading WAN contenders: Frame Relay, Integrated Services Digital Network (ISDN) and Asynchronous Transfer Mode (ATM).

Frame Relay

Frame Relay is employed to efficiently transfer information with minimal delay. Designed to handle data communication with large, bursty data packets, it uses a transmission channel only when there is data present. This offers significant advantages over LAN technologies that tie up lines for the entire duration of a session. Frame Relay’s ability to efficiently provide such on-demand transmission has garnered considerable attention. However, its inability to provide consistent performance, its imprecise specifications for congestion avoidance and recovery mechanisms, and its inefficiency in transmitting voice and video, preclude Frame Relay from being a multimedia WAN solution.

Integrated Services Digital Network

Integrated Services Digital Network (ISDN) emerged through the need for a single integrated structure that efficiently accommodated a variety of services. The most widely used network technology, ISDN has addressed the growing needs of digital data communications in a variety of industries. The ability to transport information digitally allows a single line to transmit up to 1.5 Megabits per second while providing superior quality of service over analog transmissions. Its use of digital communications and sharing of long distance lines and network control, allows ISDN to easily support wide area networking on a global scale.

The integration of workstations and telephone systems will open the door to a new generation of applications, including video teleconferencing, multimedia email, telecommuting, remote backup and servicing, and home access to databases and information services.

Asynchronous Transfer Mode

Asynchronous Transfer Mode (ATM) is the premier technology provider for real-time multimedia support and bandwidth on demand. ATM provides this functionality through call and connection management. Call management is concerned with issues such as congestion avoidance and recovery, while connection management includes considerations of link utilizations and traffic management that affect reliable and timely transmission of information. ATM's superior ability to deal with these issues, and its ability to scale down for use in LANs, is lacking in other WAN technologies. With no practical throughput barriers, ATM is scalable, and appropriate, for global networks.

The Convergence of LANs and WANs

Public carriers are planning to offer advanced networking technologies like ATM as a high-end ISDN standard strategy, eliminating protocol differences between switches used for LANs and WANs. The boundary between the LAN and the WAN will become a matter of speed and cost, not a technical hurdle.

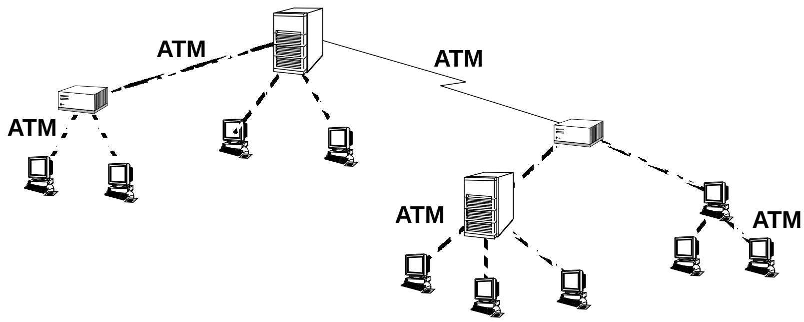

Figure 7-4 ATM can serve both local and wide-area networking requirements equally well

Key Issues in Selecting Networks

A successful network is much more than underlying transport technologies. Many factors must be considered when formulating a networking strategy. Among them are: the need for global communication, the size of business units, the number of users, and budget. Performance requirements must consider the need for real-time data access, aggregate throughput, and burst performance. Reliability concerns include uptime and redundancy options.

Distributed computing offers cost and flexibility benefits unavailable with legacy systems or PCs. However, its capabilities do translate to more sophisticated requirements for network management. Administration and diagnostic tools are needed that enable system administrators to control and test machines remotely. Routine tasks such as system configuration, security audits, and system backup should be automated, advising management personnel only when exceptions occur.

Ensuring Data Integrity

Data integrity—the ability of a system to ensure information is correct—is essential to every organization. As organizations grow, it is imperative that shared information be transmitted without error between collaborators. System administrators must ensure the network is free of faulty equipment, thereby enabling data to be transferred correctly and improving data integrity. Administrators should note that data integrity issues are generally the result of network hardware problems.

Evaluating System Performance

Users gauge application performance by the time it takes the system—both networks and servers—to process requests and fulfill the actions they request. To meet these expectations, each system on the network must have sufficient resources to handle the network traffic addressed to it.

If the network is sending packets faster than a system can receive them, packets can be dropped, requiring them to be retransmitted at least once. System administrators can use the `ping` command to determine if a system is prone to dropping packets. It sends a one-way stream of packets to a host and reports on the percentage of dropped packets, if any (Figure 7-6).

```
# ping -s seti

PING seti: 56 data bytes
64 bytes from seti (129.146.182.191): icmp_seq=0. time=0. ms
64 bytes from seti (129.146.182.191): icmp_seq=1. time=0. ms
64 bytes from seti (129.146.182.191): icmp_seq=2. time=0. ms
```

Figure 7-6 The `ping` command can be used to determine if a system is prone to losing network packets

When used with the `-s` option, the `ping` command sends one packet per second and prints one line of output for every response it receives. No output is produced if there is no response, and the offending packet is assumed to have been dropped. Round trip times are computed and packet loss statistics are generated.

More information on the `ping` command can be found in the `ping(1M)` man page.

Reducing Network Congestion

Ensuring the timely processing of requests is key to network performance. It is, therefore, critical that the network have sufficient bandwidth to handle all of the traffic generated by users. Each networking technology has a theoretical maximum bandwidth—the rate at which it can carry data. When the network becomes overcrowded, packet collisions can occur. In these situations, servers are forced to retransmit data to clients, further congesting the network and degrading overall performance.

Congestion can be reduced by dividing the network in multiple subnets that are interconnected. This has the advantage of enabling higher bandwidth interfaces, like ATM and Gigabit Ethernet, to be employed where congestion is greatest. To do so, subnets using different technologies need mechanisms that enable them to communicate with one another.

Subnets are most often connected via *gateways* and *switches*. Gateways are servers that ensure network packets get from one subnet to another. Switches are dedicated hardware solutions that enable electrical signals to be strengthened as they travel large distances. Switches pass packets to specific hosts based on a physical addressing scheme.

Figure 7-7 illustrates the use of gateways and switches in networked environments.

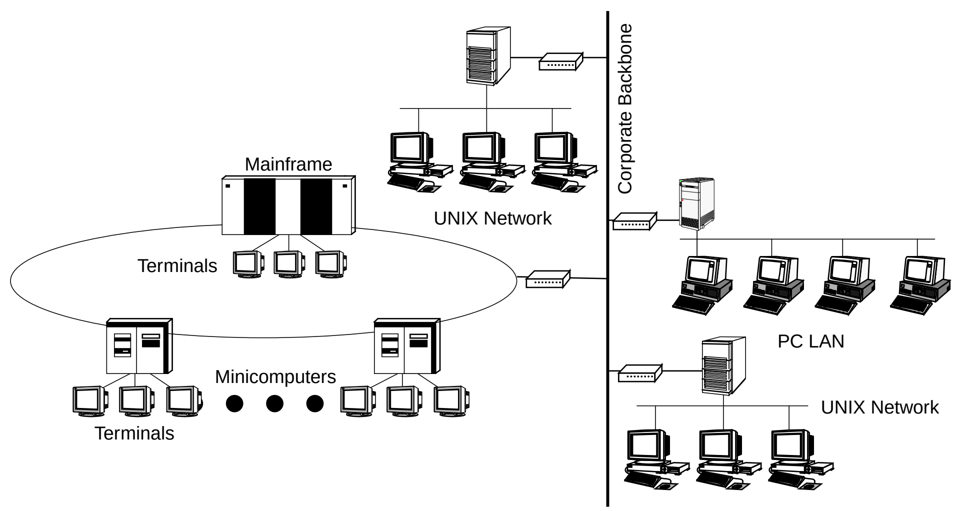

Figure 7-7 The use of gateways and switches can help reduce network congestion

NFS

The Network File System (NFS) is the industry standard distributed file system that provides transparent access to remote files and directories across a heterogeneous network. Clients no longer need copies of the files they use. Instead, files are kept on the server, reducing storage costs and aiding data consistency. Machines on the network can access the server and concurrently share files. Because NFS protocols have been widely adopted in the industry, users can access files on PCs, workstations, minicomputers, and mainframes, independent of the underlying operating systems being run.

NFS Operation

When accessing files via NFS, I/O system calls like `open()`, `close()`, `read()`, and `write()` are transformed and dissected by client software into one or more Remote Procedure Call (RPC) packets for transmission across the network. Each RPC packet represents a primitive, independent file system operation, and is referred to as an NFS operation or *NFS op*. The server runs one or more tasks to receive and process these requests. Upon the receipt of a packet, the server acts on the request, sending responses back to the client using RPC. A traditional client process requesting file service makes most of its requests synchronously — it waits for a successful completion to each request before issuing another. If a server fails to respond within a designated interval, the client will assume the original packet was lost and will issue it again.

Enhancements to NFS Version 3

For compatibility with client systems, the Netra NFS Server supports both NFS versions 2 and 3. NFS version 3 adds protocol enhancements to improve performance and reduce server load. These improvements include:

- Use of larger packet sizes to increase throughput and take advantage of high speed network technology
- Support for asynchronous writes, enabling faster writes with reduced overhead

In conjunction with NIS (Network Information Service), AutoFS, CacheFS and several other services, NFS provides the user with automatic data location, navigation and data access over wide area networks.These features all contribute to make the distribution and remote access of data fast and efficient, helping users be more productive.

The Solaris operating environment uses NFS Version 3 whenever possible, but will match clients and servers only capable of supporting NFS Version 2.

More information on using NFS efficiently can be found in *Managing NFS and NIS*, as well as the *SMCC NFS Server Performance and Tuning Guide*.

netstat

The first step toward diagnosing and correcting network congestion is understanding the current status of the network interfaces employed. System administrators can obtain the content of network related data structures using the `netstat` command.

To gauge network congestion, system administrators should pay close attention to two significant statistics generated by `netstat`:

- The number of input errors (Ierrs) and output errors (Oerrs).

  A large number of these (over 0.025%) generally indicate a hardware problem on the network.

- Collision rate (Collis)

  If the collision rate is greater than 10% of out packets (Opkts), the network is too congested.

```
# netstat -i

Name   Mtu    Net/Dest    Address     Ipkts     Ierrs  Opkts    Oerrs  Collis  Queue
lo0    8232   loopback    localhost   16860     0      16860    0      0       0
hme0   1500   nevermore   nevermore   4944832   0      261158   0      0       0
```

Figure 7-8 The use of `netstat` can help identify network congestion

More information on `netstat` can be found in *Sun Performance and Tuning, Java and the Internet, Chapter 9: Networks*, and in the `netstat(1M)` man page.

nfsstat

NFS servers are in wide use today, and have an impact on network performance. Indeed, NFS requests account for a significant portion of network traffic. Understanding the mix of NFS operations can help system administrators identify how best to tune their NFS servers and networks.

The nfsstat utility reports a variety of statistics that can be helpful in understanding the I/O mix of the NFS environment (Figure 7-5). In particular, system administrators should pay close attention to the following information:

- *calls*, the total number of RPC calls received.
- *badcalls*, the total number of RPC calls rejected.
- *nullrecv*, the number of times an RPC call was not available when it was thought to be received.
- *badlen*, the number of RPC with a length shorter than the RPC minimum.
- *xdrcall*, the number of RPC calls whose header could not be decoded.
- *readlink*, the number of links read. If greater than 10% of total I/O mix, the link should be replaced with a directory.
- *getattr*, the number of times file attribute information was requested. If greater than 60% of the total I/O mix, ensure that the attribute cache value of NFS clients is set appropriately.
- *null*, if more than 1 percent, automounter time-out values are set too short.
- *writes*, if more than 5 percent, a caching mechanism like PrestoServe or NVRAM should be configured into the server.

More information on the use and interpretation of `nfsstat` can be found in the *Solaris 2.4 SMCC NFS Server Performance and Tuning Guide*, as well as the `nfsstat(1M)` man page.

snoop

Identifying network traffic patterns and determining their cause can be a time-consuming task. The `snoop` command aids this effort, enabling network packets to be captured and analyzed. Using both network packet filters and streams buffer modules, the `snoop` command enables the efficient capture of packets from then network.

More information on the `snoop` command can be found in the `snoop(1M)` man page.

PrestoServe

PrestoServe is a Solaris device driver that uses non-volatile memory to cache write requests. File system and I/O requests are sent to the PrestoServe device driver. Write data is cached, and read data is returned, if available. When PrestoServe needs to perform I/O to the device, it issues commands via standard device driver routines.

The PrestoServe device driver improves performance by reducing latency — all data is written to a cache in non-volatile memory rather than to a much slower disk drive. It also reduces the total amount of disk I/O by cancelling reads and writes that overlap data already in the cache.

PrestoServe and Fast NFS Writes

In NFS environments like the Netra NFS Server, one instance of PrestoServe is layered above the storage management software and below the UFS file system. Each NFS operation that goes through the file system can benefit from the PrestoServe cache. This instance of PrestoServe, called *PrestoServe Upper*, significantly improves NFS response time and write throughput.

PrestoServe and Fast RAID-5 Writes

Some NFS environments, like the Netra NFS Server, use Solstice DiskSuite to manage RAID-5 disk subsystems. Each RAID-5 write request translates into six disk I/O operations: two operations to read the old data and parity, two operations to write the new data and parity to a pre-write log, and two operations to write the new data and parity to its actual disk location. The pre-write log ensures reliability and prevents data corruption in the event of a system crash during a write operation.

Clearly, synchronous write operations are I/O intensive and have an adverse effect on NFS throughput and response time. PrestoServe dramatically improves synchronous write performance by eliminating six disk I/O operations each time the PrestoServe Upper cache is utilized.

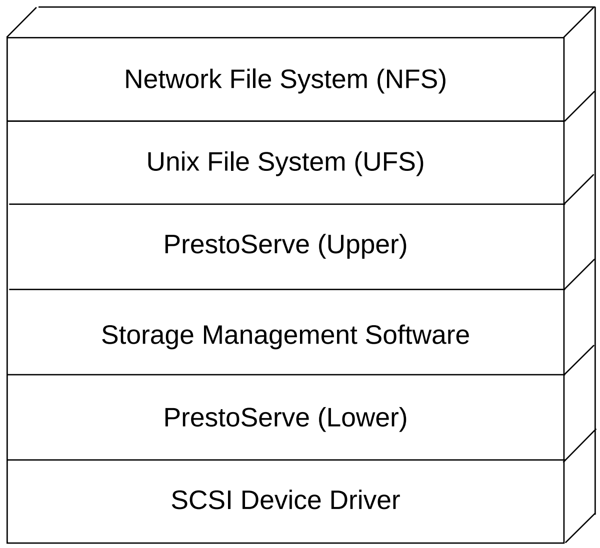

Figure 7-9 Some NFS environments use two instances of PrestoServe in conjunction with storage management software to manage disk subsystems and improve I/O performance

Systems can also utilize a second instance of PrestoServe, called *PrestoServe Lower*, to cache writes to the RAID-5 pre-write area (Figure 7-6). Stable memory is used to cache writes to a pre-write log, eliminating two disk I/O operations and significantly improving performance. Since the pre-write log is in a fixed location and is relatively small, 99 percent of pre-writes are eliminated. The use of a RAID-5 pre-write area generates two I/O operations for each write. By cancelling these, PrestoServe Lower reduces the number of disk I/O operations from six to four for each RAID-5 write.

More information on PrestoServe as implemented in the Netra NFS Server can be found in *The Netra NFS Server* technical white paper located at **xxx** on the World Wide Web.

CacheFS

The caching file system in Solaris can help shield Solaris client systems from NFS server and network outages. CacheFS transparently maintains a local copy of all recently-used NFS data in the client system's disk. As most user I/O activity is data reading, cacheFS uses the network and server much less frequently than NFS or other file-sharing systems. Consequently, cacheFS users are less likely to notice temporary network or NFS server outages. This feature will be enhanced in the future to extend the interval during which users can be shielded from outages.

More information on CacheFS can be found in *Sun Performance and Tuning, Java and the Internet, Chapter 7: Applications*.

WebNFS

The World Wide Web is rapidly becoming a new computing platform. Originally, users were unable to access files across the Internet. Today, WebNFS provides what the Web needs most — a file system that enables collaboration, connection management and concurrency while ensuring the performance and scalability of NFS operations over the network. With WebNFS, users are no longer limited to viewing documents. They may now read and write documents over the corporate intranet or the Internet.

WebNFS is an enhanced version of Sun's NFS network file system protocol. It is compatible with firewalls and provides a global naming scheme that allows Web-enabled applications to transparently access data while offering remarkable performance gains. WebNFS complements existing HTTP and FTP protocols while providing faster, more robust, and scalable file transfers.

WebNFS is an industry-leading technology that offers a host of benefits:

- *Web-based collaboration*

  WebNFS allows Web data to be easily integrated into existing applications. Today, most users access data through a browser, forcing them to cut and paste information into their local applications. WebNFS can be used to Web-enable any desktop application by providing file access methods that are compatible with the way applications typically access local disks.

- *Connection management*

  With WebNFS, clients can download multiple files over a single, persistent TCP connection, reducing network congestion and speeding overall throughput.

- *Concurrency*

  WebNFS clients can issue multiple, concurrent requests to an NFS server, resulting in more effective server and network utilization and better performance.

- *Fault tolerance*

  WebNFS is well known for its fault tolerance in the event of network and server failures. When interrupted, other file transfer protocols require downloads to be restarted. However, WebNFS can resume a download from where it left off, reducing duplicate effort and minimizing overall transfer time.

- *Streamlined access*

  Because it bypasses the *mount* protocol, WebNFS provides significantly faster file downloads than standard NFS implementations. In addition, the ability to support partial file transfers saves time when only a portion of a large file is needed.

More information on WebNFS can be found in the WebNFS Administration Tasks book located at *http://docs.sun.com* on the World Wide Web, as well as the *WebNFS — The Filesystem for the Internet* white paper located at *http://www.sun.com/webnfs/wp-webnfs/*.

Summary of Guidelines for System Performance Evaluation 7

System tuning can be a difficult process. Locating bottlenecks and identifying their root cause can be time consuming. Five major categories must be monitored, evaluated, and tuned:

- Overall system usage
- CPU resources
- Memory resources
- Disk subsystems
- Network resources
- NFS activity

System administrators can take advantage of virtual_adrian.se, a personalized performance monitoring and tuning tool. When the system boots, virtual_adrian.se performs some simple tuning, and every 30 seconds examines the state of the system. Users are notified only when a problem occurs, and displays the data causes the issue. Administrators can then take appropriate actions to rectify the problem based on suggestions made by the tool. More information on the virtual_adrian.se tool can be found in *Sun Performance and Tuning, Java and the Internet, Chapter 16: The SymbEL Example Tools.*

If the virtual_adrian.se tool is not available, administrators can analyze the system directly. The following sections are intended to provide a set of first steps to be taken when attempting to diagnose, detect, isolate, and fix performance bottlenecks, and are best performed in the order written.

System Usage

The first step toward understanding the performance of a system lies in the determination of overall system usage. To check system usage, administrators should use the `vmstat` command (Figure A-1).

```
# vmstat -S 5
```

Figure A-1 The first step in evaluating performance bottlenecks in determining overall system usage.

If the CPU is not saturated (id > 15%), then adding more processors will not solve the problem. However, CPU utilization should be checked.

Next, administrators should check memory usage. If there is sufficient memory (po > 0 or so > 0) then additional memory will not be a factor in improved performance, and the disk subsystem is the next place to check for bottlenecks. If there is insufficient memory, then the following steps can be taken:

- Add memory to the system
- Reconfigure the Solaris operating environment kernel
- Redistribute swap space
- Tune the paging algorithm
- Rearrange the process load
- Use shared libraries
- Set memory limits

Figure A-2 summarizes these steps in flowchart form.

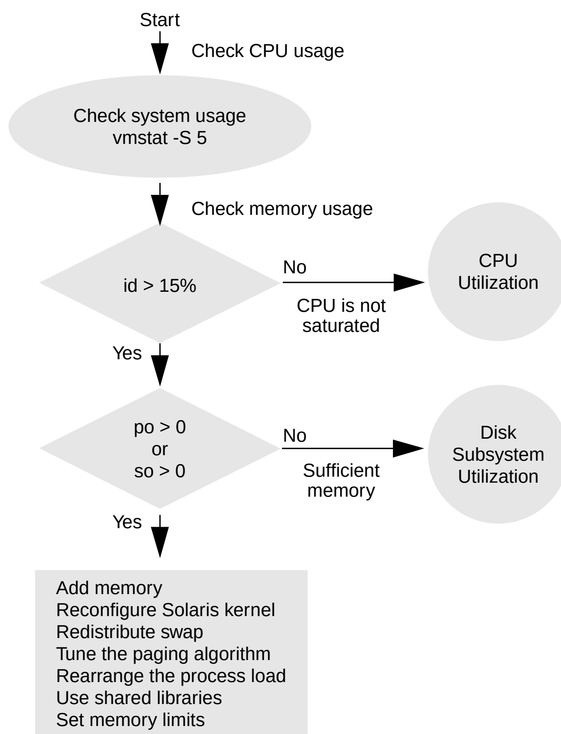

Figure A-2 The high level steps in locating bottlenecks and tuning systems for high performance.

CPU Utilization

Heavy system and user CPU usage can be caused by a variety of conditions, and the evaluation of CPU utilization can lead to a number of determinations. It is possible that sufficient resources exist but are experiencing hardware failures. Another possibility is that a large number of processes are vying for a limited number of resources. In these scenarios, upgrading to a faster processor or to multiple processors can help the situation and improve system performance.

The first step in evaluating CPU utilization is to determine whether there is heavy use of the CPU by the system or by users. System administrators should first run the `vmstat` command. If system CPU usage is greater than 30 percent, then the system is under heavy system CPU utilization. In this case, if the number of interrupts is high, then a hardware problem may exist. Otherwise a disk or network bottleneck may be impacting system performance.

If the system is experiencing high levels of user CPU usage, then a decision must made concerning the number and type of processors in the system. If there are processes contending for the CPU, then upgrading to a multiprocessor system is in order. Otherwise, upgrading to a faster CPU will generally mitigate the performance degradation being experienced. Figure A-3 summarizes these steps in flowchart form.

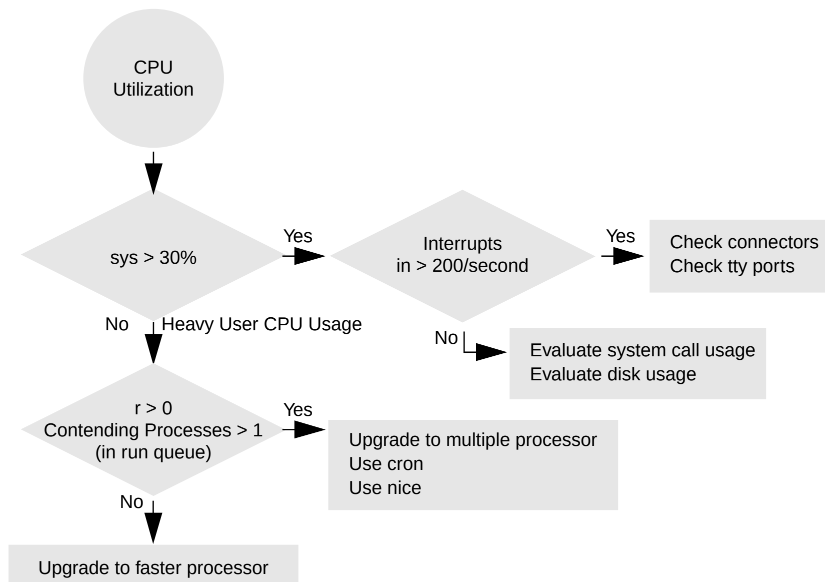

Figure A-3 The steps to evaluating CPU utilization.

Disk Subsystem Utilization

The disk subsystem is often the slowest component of any computing platform. With datasets growing by as much as 100 percent per year, ensuring the disk is not a bottleneck is a critical factor to overall system performance. System administrators need to monitor disk usage and determine if the system is experiencing heavy write activity, unbalanced disk usage, or has saturated one or more disk subsystems.

Two utilities can be used to monitor disk usage: `iostat` and `vmstat`. System administrators should first run iostat. If there is heavy write activity (writes/second > reads/second), then a caching mechanism like PrestoServe or 32 MBNVRAM board should be added to the system. If, however, any of the disks are busy more than 60 percent of the time, then they should be considered saturated and their data distributed onto additional spindles. Lastly, if the busy percentage of the disks on the system is uneven by more than 20 percent, the data should be relocated to additional disks and RAID 0 (striping) used to help I/O performance.

Once the results of `iostat` have been evaluated and analyzed, system administrators can use `vmstat` to check for other conditions. If `vmstat` reports light disk usage, then administrators should turn their attention to the network components of the system. If, on the other hand, heavy disk usage is observed, three conditions should be analyzed: the number of users, the amount of memory, and overall system capacity.

If `vmstat` indicates that the system is frequently busy retrieving entries from the Directory Name Lookup Cache (DNLC), system administrators can increase the MAXUSERS and MAX_NPROC environment variables used by Solaris, resulting in a larger name cache being employed. If there is sufficient free memory on the system but the use of temporary files is great, then the tmpfs file system can be added to the system to increase caching and improve read/write performance. Finally, if the disk capacity has been exceeded, more disks need to be added to the system and the load redistributed among all disks and file systems.

Figure A-4 summarizes these steps in flowchart form.

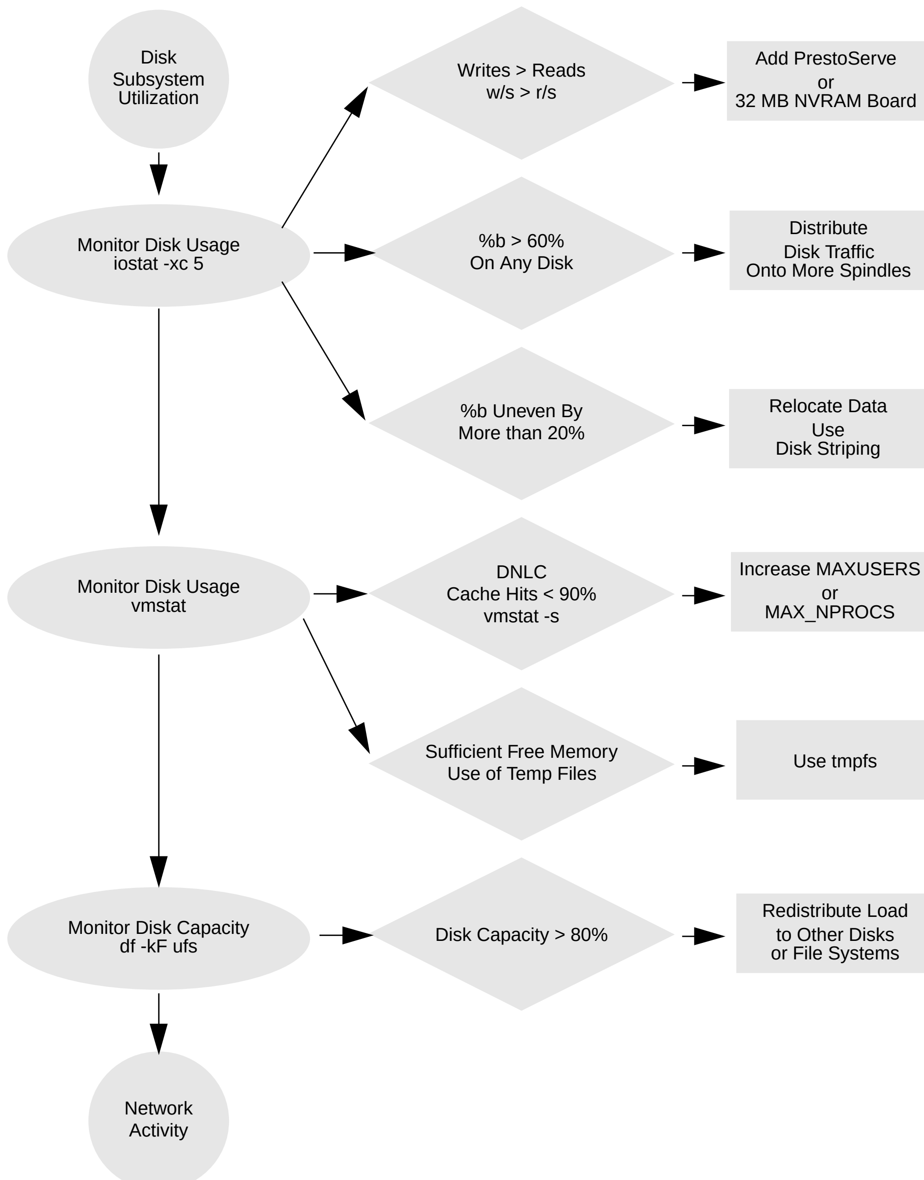

Figure A-4 The steps to evaluating disk utilization.

Network Activity

Networks are a critical component of the enterprise infrastructure. If networks do not perform well, collaboration and communication can suffer. System administrators need to monitor network activity and determine if the network is congested, a hardware problem exists, or if some network parameters and components need to be reconfigured.

The `netstat` utility can be used to monitor network activity. System administrators should first run `netstat` with the `-i` option to determine if the network is saturated, broken, or poorly configured.

- If `netstat` indicates the network is saturated (Collisions/Opkts > 5%), the network should be divided into subnets and faster interfaces, like Gigabit Ethernet, employed.
- If `netstat` indicates a network hardware problem (Oerrs/Opkts > 0.025%), all hardware components should be checked and replaced, if necessary.
- If, however, there are insufficient receive buffers (Ierrs/Ipkts > 0.025%), system administrators should increase the number of receive buffers and reconfigure the network.

System administrators should run `netstat` a second time, using the `-s` option. If `netstat` indicates that there are insufficient NFS daemon running to handle the workload, then administrators should increase the number of daemons to handle the UDP socket overflow problem that typically results in these situations.

Figure A-5 summarizes these steps in flowchart form.

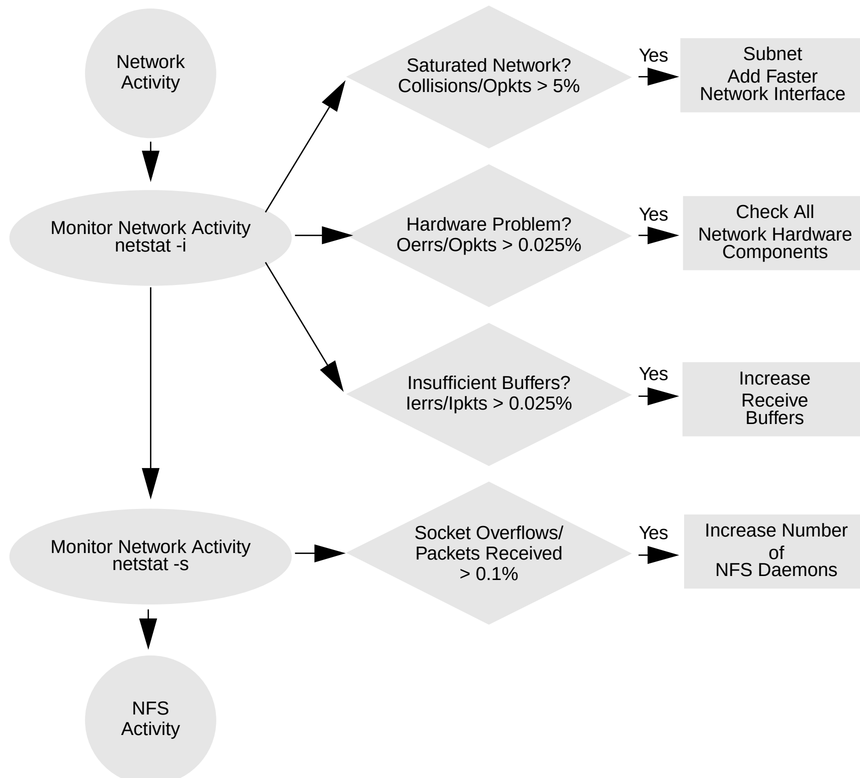

Figure A-5 The steps to evaluating network activity.

NFS Activity

The final step in analyzing network performance is determining the effect of NFS activity on the system. The `nfsstat` utility is available for this purpose. System administrators should use `netstat` with the `-rc` option to isolate network and server bottlenecks. The `-s` option should also be used, enabling administrators to determine if file system caches are too small, or if NFS writes need to be cached.

System administrators should first run `netstat` using the -rc option and determine if network and server bottlenecks are hampering NFS activity:

- If there is a network bottleneck, the ratio of retransmissions to calls will be high (retrans/calls > 5%). Administrators should check the connections between network devices, and decrease the size of reads and writes.

- If there is a server bottleneck, timeouts should be increased and the server potentially upgraded.

System administrators should then run `netstat` with the `-s` option and determine if the caching mechanisms are properly configured.

- If the file system inode cache is too small (getattr + setattr > 50%), system administrators should increase the MAXUSERS environment variable.
- If users are making extensive use of symbolic links (readlink > 10%), administrators should replace the most heavily used symbolic links with mount points on the client.
- If the system is experiencing a high degree of NFS writes, a caching mechanism such as PrestoServe or 32 MB NVRAM board should be added to the server.

Figure A-6 summarizes these steps in flowchart form.

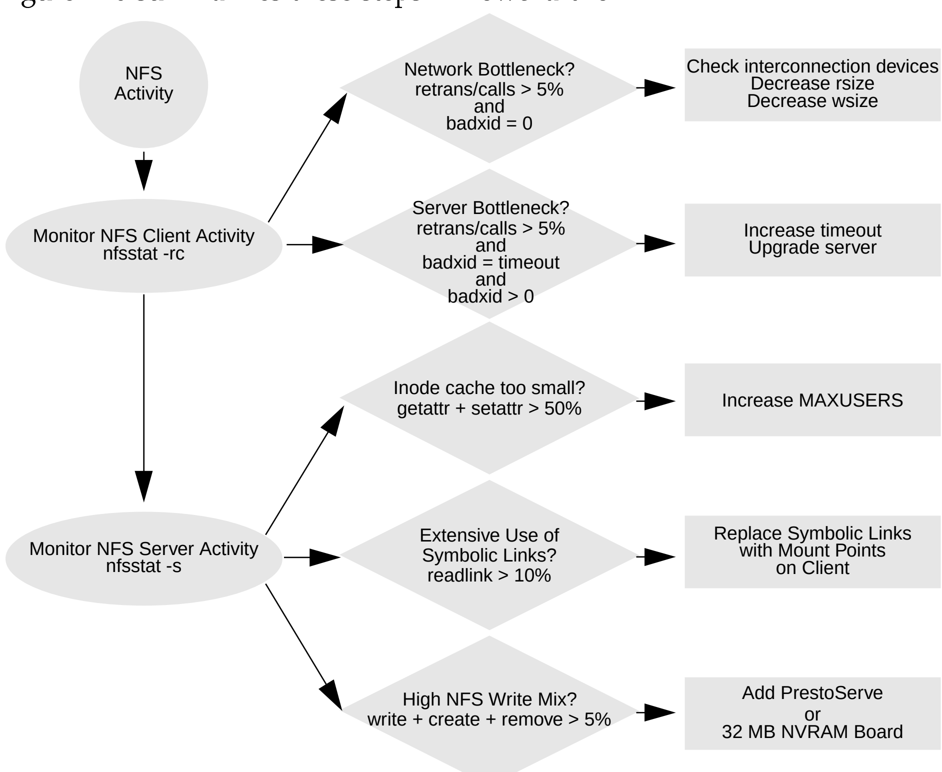

Figure A-6 The steps to evaluating NFS activity.

Glossary

A

Arbitrated loop

A loop technology in which two or more ports can be interconnected, but only two ports at a time may communicate.

Asynchronous Transfer Mode

A networking protocol based on cell-switching technology that offers high throughput.

ATM

Asynchronous Transfer Mode.

Availability

A measure of the total time that data is available from a system.

Channel

An interface directed toward high speed transfer of large amounts of information.

Concatenation

A volume created by sequentially mapping blocks on disks to a logical device. Two or more partitions can be concatenated and accessed as a single device.

Disk array

A subsystem that contains multiple disk drives, designed to provide performance, high availability, serviceability, or other benefits.

Disk group

A grouping of disk drives and the data on them that facilitates organization and the movement of disks between systems.

Ethernet

A networking technology that provides 10 Megabit/second transfer rates for Local Area Networks.

Fabric

A group of interconnections between ports that includes a hub, switch, or arbitrated loop.

Fast Ethernet

An extension of Ethernet that provides 100 Megabit/second network transfer rates.

Fiber

A wire or optical strand. Spelled "fibre" in the Fibre Channel name.

Fiber-optic cable

Jacketed cable container of thin strands of glass through which pulses of light transmit data. Used for high speed transmission over medium to long distances.

Frame

An indivisible unit for transfer of information in Fibre Channel networks.

Full duplex

A communications protocol that permits simultaneous transmission in both directions, usually with flow control.

Gigabit Ethernet

A networking technology that provides 1000 Megabit/second transfer rates for networks.

GUI

Graphical User Interface.

Hot relocation

A Sun Enterprise Volume Manager feature in which data is automatically reconstructed on a spare disk after a disk failure without interruption to user access.

Hot Spare

A drive in an array that is held in reserve to replace any other drive that fails. Hot spares are continuously powered up and spinning. This allows the array processor to have immediate access to a functioning drive for possible reconstruction of lost data.

Hot Swap

The ability to remove and replace drives without affecting system operation.

Hub

A device used to connect network cables.

Integrated Services Digital Network

A low cost wide area networking technology that is rapidly increasing in popularity. Using digital signalling over conventional telephone lines, ISDN can provide up to 128 KB/second transfer rates, with even higher rates defined for broadband transmission media.

I/O rate

A measure of the capacity of a device to transfer data to and from another device within a given time period; typically stated in I/O operations per second.

IOPS

Input/output operations per second. A measure of I/O performance, this is commonly used to quote random I/O performance.

IP

Internet Protocol. A set of protocols developed by the United States Department of Defense to communicate between dissimilar computers across networks.

ISDN

Integrated Services Digital Network.

LAN

Local Area Network.

Laser

Light Amplification by Stimulated Emission of Radiation. A device for generating coherent radiation in the visible, ultraviolet, and infrared portions of the electromagnetic spectrum. Used as an illumination source in fiber-based communications.

Link

One inbound fiber and one outbound fiber connected to a port.

Local Area Network

A network topology that provides a means to connect systems within a limited distance.

Micron

One millionth of a meter. Also called micrometer.

Mirroring

In RAID terminology, refers to the redundant storage of data by duplicating the data.

Multimode fiber

An optical fiber which allows light to travel along multiple paths, resulting in gradual signal degradation with distance.

Network

An arrangement of nodes and connecting branches, or a configuration of data processing devices and software connected for information exchange.

Network File System

A data management facility that allows remote files and machines to appear local.

NFS

Network File System.

Optical fiber

Any filament of fiber, made of dielectric material, that guides light.

Parity

In an array environment, data that is generated from user data and is used to regenerate user data lost due to a drive failure. Used in RAID 3 and RAID 5.

Point-to-point

A topology in which exactly two ports communicate at a time.

Port

An access point in a device where a link attaches.

Protocol

A convention for data transmission that defines the sequencing of information transfers.

RAID

Redundant Array of Independent (or Inexpensive) Disks. A set of disk drives that appear to be a single logical disk drive to an application such as a database or file system. Different RAID levels provide different capacity, performance, availability, and cost characteristics.

RAID-0

RAID level 0, or striping. Data is distributed among disks for performance. No redundancy is provided, and the loss of a single disk causes the loss of data on all disks in the stripe.

RAID 0+1

The combination of striping and mirroring. Data is distributed among disks for performance, and mirroring is used to provide redundancy.

RAID-1

RAID level 1, or mirroring. Multiple copies of the data are kept. This is inherently expensive because 100% duplication of data is required, or 200% in the case of a three-way mirror.

RAID-5

RAID level 5, or striping with distributed parity. Both data and parity are distributed across disks. No single disk can compromise the integrity of the data. RAID-5 optimizes performance, reliability and cost.

Redundancy

Duplication for the purpose of achieving fault tolerance. Refers to duplication or addition of components, data and functions within the array.

SCSI

Small Computer Systems Interface. An ANSI standard for controlling peripheral devices by one or more host computers.

Serial transmission

A data communications mode in which bits are sent in sequence through a single signal path.

Single-mode fiber

A step index optical fiber in which light propagates coherently, avoiding signal degradation by distance.

Snapshot

A Sun Enterprise Volume Manager feature for enabling on-line backups. Snapshots are read-only copies of the data to be backed up.

Striping

Spreading, or interleaving, logical contiguous blocks of data across multiple independent disk spindles. Striping allows multiple disk controllers to simultaneously access data, improving performance.

Switch

A networking device that isolates network traffic to the segments and devices for which the traffic is intended.

TCP/IP

Transport Control Protocol/Internet Protocol, a DARPA-defined network protocol suite featuring a connection-less network layer.

Throughput

A measure of sequential I/O performance, quoted as MB/second. See IOPS and I/O rate.

Topology

The components used to connect two or more ports together. Also, a specific scheme of connecting those components. Point-to-point, fabric, and arbitrated loop are example topologies.

Transceiver

A transmitter/receiver module.

Transfer rate

The rate at which data is transferred. Usually measured in MB/second.

Volume

A volume is a virtual disk into which a file system, DBMS, or other application can place data. A volume can physically be a single disk partition or multiple disk partitions on one or more physical disk drives. Applications that use volumes do not need to be aware of their underlying physical structure. Software handles the mapping of virtual partition addresses to physical addresses.

WAN

Wide Area Network.

Wide Area Network

A network topology that provides a means to connect systems which are distributed over a large geographic area.

References

Sun Microsystems posts product information in the form of data sheets, specifications, and white papers on its Internet World Wide Web Home page at: http://www.sun.com.

Look for abstracts on these and other Sun technology white papers and manuals:

Delivering Performance on Sun: Tools for High Performance Distributed Computing, Sun Microsystems, 1998.

Delivering Performance on Sun: Optimizing Applications for Solaris, Sun Microsystems, 1998.

Fibre Channel Technology, Technical Brief, Sun Microsystems, 1994.

The microSPARC-II Processor, Technology White Paper, Sun Microsystems, 1995.

Multiprocessing WorkstationTechnology fromSun Microsystems, Sun Microsystems, 1998.

The Netra NFS Server, Technical White Paper, Sun Microsystems, 1997.

Networking and Sun, Where the Network is Going..., Sun Microsystems, 1995.

Networking the Enterprise, Sun's Vision for Network Computing, Sun Microsystems.

The NFS Distributed File Service, Sun Microsystems, 1995. Available at *http://www.sun.com/solaris/wp-nfs*.

Reliability, Availability, and Serviceability in the Sun StorEdge A5000 Disk Array, Sun Microsystems, 1998.

Scalable Data Mining with Sun Ultra Enterprise Servers, Sun Microsystems, January 1997.

Sun Performance and Tuning, Java and the Internet, Adrian Cockcroft, Sunsoft Press, 1998.

The Sun StorEdge A5000 Architecture, Technical White Paper, Sun Microsystems, 1998.

Sun Solaris Operating Environment, Sun Microsystems, June 1997, available at *http://www.sun.com/solaris/wp-solaris2.6*.

The TurboSPARC Processor, Technology White Paper, Sun Microsystems, 1997.

The Ultra 5 and Ultra 10 Workstation Architecture, Sun Microsystems, 1998.

The UltraSPARC Processor, Technology White Paper, Sun Microsystems, 1997.

The UltraSPARC-IIi Processor, Technology White Paper, Sun Microsystems, 1998.

Web sites of interest:

`http://access1.sun.com/workshop`

`http://docs.sun.com`

`http://www.sun.com/software/whitepapers.html`

`http://www.sun.com/sparc/vis`

`http://www.sun.com/sparc/whitepapers`

`http://www.sun.com/workshop/sitemap.html`

`http://www.sun.com/workshop/workshopFAQ.html`

Sun Microsystems Computer Company
A Sun Microsystems, Inc. Business
901 San Antonio Road
Palo Alto, CA 94303 USA
415 960-1300
FAX 415 969-9131
http://www.sun.com

Sales Offices
Argentina: +54-1-311-0700
Australia: +61-2-9844-5000
Austria: +43-1-60563-0
Belgium: +32-2-716-7911
Brazil: +55-11-524-8988
Canada: +905-477-6745
Chile: +56-2-638-6364
Colombia: +571-622-1717
Commonwealth of Independent States:
+7-502-935-8411
Czech/Slovak Republics:
+42-2-205-102-33
Denmark: +45-44-89-49-89
Estonia: +372-6-308-900
Finland: +358-0-525-561
France: +33-01-30-67-50-00
Germany: +49-89-46008-0
Greece: +30-1-680-6676
Hong Kong: +852-2802-4188
Hungary: +36-1-202-4415
Iceland: +354-563-3010
India: +91-80-559-9595
Ireland: +353-1-8055-666
Israel: +972-9-956-9250
Italy: +39-39-60551
Japan: +81-3-5717-5000
Korea: +822-3469-0114
Latin America/Caribbean:
+1-415-688-9464
Latvia: +371-755-11-33
Lithuania: +370-729-8468
Luxembourg: +352-491-1331
Malaysia: +603-264-9988
Mexico: +52-5-258-6100
Netherlands: +31-33-450-1234
New Zealand: +64-4-499-2344
Norway: +47-2218-5800
People's Republic of China:
Beijing: +86-10-6849-2828
Chengdu: +86-28-678-0121
Guangzhou: +86-20-8777-9913
Shanghai: +86-21-6247-4068
Poland: +48-22-658-4535
Portugal: +351-1-412-7710
Russia: +7-502-935-8411
Singapore: +65-224-3388
South Africa: +2711-805-4305
Spain: +34-1-596-9900
Sweden: +46-8-623-90-00
Switzerland: +41-1-825-7111
Taiwan: +886-2-514-0567
Thailand: +662-636-1555
Turkey: +90-212-236-3300
United Arab Emirates:
+971-4-366-333
United Kingdom: +44-1-276-20444
United States: +1-800-821-4643
Venezuela: +58-2-286-1044
Worldwide Headquarters:
+1-415-960-1300

Printed in USA